Decoding holographic codes with an integer optimization decoder

Robert J. Harris,[1,*] Elliot Coupe,[1] Nathan A. McMahon,[1,†] Gavin K. Brennen,[2] and Thomas M. Stace[1]

[1]*ARC Centre for Engineered Quantum Systems, School of Mathematics and Physics, The University of Queensland, St. Lucia, Queensland 4072, Australia*

[2]*Center for Engineered Quantum Systems, Department of Physics and Astronomy, Macquarie University, Sydney, New South Wales 2109, Australia*

(Received 24 August 2020; revised 29 November 2020; accepted 30 November 2020; published 21 December 2020)

We develop a most likely error Pauli error decoding algorithm for stabilizer codes based on general purpose integer optimization. Using this decoder we analyze the performance of holographic codes against Pauli errors and find numerical evidence for thresholds against Pauli errors for bulk qubits. We compare the performance of holographic code families of various code rates and find phenomenological Pauli error thresholds ranging from 7% to 16%, depending on the code rate. Additionally we give numerical evidence that specific distance measures of the codes we consider scale polynomially with number of physical qubits.

DOI: 10.1103/PhysRevA.102.062417

I. INTRODUCTION

Recent interest in the overlap between quantum information and the bulk-boundary correspondence [1] has connected holography and error correction [2–4]. In holography, degrees of freedom (DOFs) in a $d+1$ dimensional bulk manifold are encoded in DOFs in a d dimensional boundary manifold, leaving substantial redundancy to the encoding. This redundancy makes holography a principle for developing quantum error correcting codes.

For practical quantum processing, error correction is essential [5,6]. A good quantum error correcting code (QECC) ideally has four properties: a threshold against physical errors [7], a finite ratio—the code rate—of the number of encoded to physical qubits [5], low-weight parity check operators [8], and a fault-tolerant path to a universal gate set [9]. In this paper, we study code thresholds, rates, and distances for a variety of holographic code families, to establish their performance against some of these metrics.

Pastawski *et al.* [3] (HaPPY) developed holographic quantum error correcting codes by tessellating tensor representations of the five-qubit QECC [10] on discretized negatively curved space. Due to the proportional scaling between bulk and boundary volumes in a hyperbolic tiling, holographic error correcting codes can form finite rate error correcting codes. The rate of the code is tunable via modification of the tiling. HaPPY showed that the code with the maximum achievable rate did not have thresholds against erasures; however, a version of the code with reduced rate was shown to have an erasure threshold which is comparable to that of the surface code [7,11,12].

The five-qubit QECC was the first code used to generate holographic tilings because it has a *perfect tensor* representation. This property is rare amongst known codes. More recently it has been shown that the perfect tensor condition can be relaxed to a *block perfect* condition [13], which coincides with the definition of planar perfect tangles [14]. This relaxation allows more choices for the seed codes used to generate the holographic tiling, extending the set of known holographic codes families to include a holographic code based on the Steane code [15] (called here the *heptagon code* [13]). This was also shown to have promising erasure thresholds [13].

Calderbank-Shor-Steane (CSS) codes have a transversal CNOT gate; self-dual CSS codes also have transversal Hadamard gates [16]. Additionally it is known how to construct any CSS code with graph states [17,18], which is a powerful approach to measurement-based quantum computation.

Decoding erasures is numerically straightforward [13], but optimally decoding computational (e.g., Pauli) errors is a numerically hard problem in general [19] (though good decoders exist for specific codes [20–22], such as the surface code [23–26] and the color code [27,28]). Given the relative computational ease of decoding erasure errors, the performance of a new QECC against erasure has been proposed as a performance filter [24,25]. The promising erasure thresholds for holographic codes therefore suggests further study is warranted.

In this paper we describe a general purpose decoding algorithm for stabilizer codes based on a global branch and bound integer optimization [29]. We implement this decoder and numerically compute the performance of various holographic code families, including a holographic code family based on a small surface code that we introduce here. We provide numerical evidence of thresholds against Pauli errors for these holographic code families. Additionally we use related algorithms to find distances of the codes.

II. HOLOGRAPHIC CODE CONSTRUCTION

Holographic codes are seeded by an $[n,k,d]$ QECC [30]. The seed QECC is described by a rank $(n+k)$ tensor, T,

*rjh2608@gmail.com

†Present address: Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Institute of Theoretical Physics, Erlangen, Germany.

TABLE I. The ordered set of *extended* stabilizers and logical operators that define the block-perfect Steane code tensor, $T_S^{j_1,\ldots,j_L,j_7}$. The tensor is given by $T_S^{j_1,\ldots,j_L,j_7} = \langle j_1,\ldots,j_L,j_7|T_S\rangle$, where $|T_S\rangle$ is the $+1$ eigenstate of the eight commuting operators, S_j.

Index label:	1	2	3	4	5	6	L	7	
	X	X	$\mathbb{I}$	$\mathbb{I}$	$\mathbb{I}$	X	$\mathbb{I}$	X	$\equiv S_1$
	$\mathbb{I}$	X	X	X	$\mathbb{I}$	$\mathbb{I}$	$\mathbb{I}$	X	$\equiv S_2$
	$\mathbb{I}$	$\mathbb{I}$	$\mathbb{I}$	X	X	X	$\mathbb{I}$	X	$\equiv S_3$
	Z	Z	$\mathbb{I}$	$\mathbb{I}$	$\mathbb{I}$	Z	$\mathbb{I}$	Z	$\equiv S_4$
	$\mathbb{I}$	Z	Z	Z	$\mathbb{I}$	$\mathbb{I}$	$\mathbb{I}$	Z	$\equiv S_5$
	$\mathbb{I}$	$\mathbb{I}$	$\mathbb{I}$	Z	Z	Z	$\mathbb{I}$	Z	$\equiv S_6$
	X	X	X	X	X	X	X	X	$\equiv S_{\bar{X}}$
	Z	Z	Z	Z	Z	Z	Z	Z	$\equiv S_{\bar{Z}}$

which is the encoding map from logical to physical space $T : \mathcal{H}_k \mapsto \mathcal{H}_n$. To be a valid code, T must be an isometry, $T^\dagger T \propto \mathbb{I}$, which ensures perfect information recovery in the absence of noise. We define the order-$(n+k)$ seed tensor as $T^{j_1,\ldots,j_{n+k}} = \langle j_1,\ldots,j_{n+k}|T\rangle$, where $|T\rangle$ is the $+1$ eigenstate of all *extended* stabilizers and logical operators, i.e., the extension of the operators to act on an $(n+k)$-qubit Hilbert space.

Conventionally, the tensor indices are arranged to form an encoding map, $T^{j_1,\ldots,j_n \leftarrow j_{n+1},\ldots,j_{n+k}} : \mathcal{H}_k \mapsto \mathcal{H}_n$, from the k dimensional logical Hilbert space to the n dimensional physical Hilbert space. Related codes can be defined with different index partitions, providing the encoding tensor remains an isometry. For example, if $T^{j_2,\ldots,j_n \leftarrow j_1,j_{n+1},\ldots,j_{n+k}}$ is an isometry, then it is an encoding map for a $[n-1, k+1, d']$ code.

III. PERFECT AND BLOCK-PERFECT TENSORS

Some classes of tensor remain an isometry after a permutation of indices. We describe two such classes. Take a tensor with m indices in an ordered set $J = \{j_1, j_2, \ldots, j_m\}$. Partition J into ordered subset A and its complement $\overline{A}$, with some permutation Π with respect to the reference index, i.e., $\{A|\overline{A}\} = \Pi[J]$. If under all permutations Π the tensor remains an isometry, then it is known as a *perfect tensor* [3]. If the tensor remains an isometry under all cyclic permutations $\Pi = \sigma^p$, where $\sigma^p : j_i \to j_{i+p}$, we call it *block perfect*.

As an example, the order-8 tensor associated with the Steane code [15], $T_S^{j_1,\ldots,j_L,j_7}$, is block perfect [13] with respect to the ordered set of extended operators defined in Table I. The code family based on this seed code will be used in the following numerical results.

We also introduce a holographic code based on the $[5, 1, 2]$ *surface code fragment* (SCF), defined over five qubits, shown in Fig. 1(a). The SCF is a block-perfect, error-detecting CSS code. The order-6 SCF tensor $T_{\text{SCF}}^{j_1,\ldots,j_L,j_5}$ defined by the stabilizers in Table II is an isometry for any block partition of indices from any cyclic permutation of $\{1, 2, 3, 4, L, 5\}$.

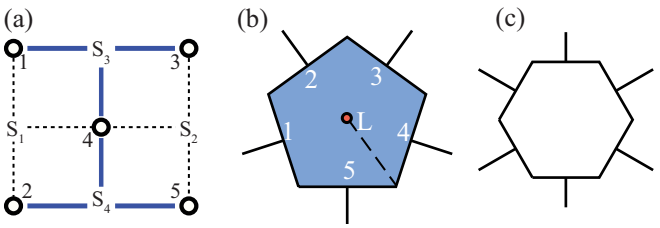

FIG. 1. (a) Lattice representing the $[5, 1, 2]$ surface code fragment (SCF) including stabilizers. Graphical representations of (b) a $[5, 1]$ code tensor and (c) the $[6, 0]$ state associated to a $[5, 1]$ code.

Seed code tensors can be represented graphically as a polygon with a leg for each index of the tensor [31]. For a $[n, k, d]$ error correcting code, the standard graphical representation is an n-sided polygon, with a leg on each edge for the physical qubit indices and k *bulk* legs (dots) perpendicular to the face representing each logical qubit. For example, Fig. 1(b) is a representation of a $[5, 1]$ code.

We conventionally interpret polygons like Fig. 1(b) as a map from bulk to planar indices. However, tensor indices can be partitioned into other subsets A and $\overline{A}$: if the tensor is an isometry for a particular choice of indices A, it describes an encoding map from A to $\overline{A}$. For example, if the tensor in Fig. 1(b) is block perfect, then the tensor will be an isometry from the indices $\{L, 5\}$ to $\{1, 2, 3, 4\}$, corresponding to a $[4, 2]$ code.

To create holographic codes we tessellate seed codes, represented graphically as polygons with legs, on a discretized 2D hyperbolic space, and contract tensor indices on linked edges. The tessellation is truncated at a particular *radius*, R. For the heptagon code [13] we tessellate heptagons which individually represent the Steane code [15]; the version for $R = 3$ is shown in Fig. 2(a). If every polygon in the tessellation has an associated logical qubit, we call it a *max-rate* holographic code.

Additionally, we consider some tensor networks in which a subset of tiles in the tessellation do not have a logical input, as depicted in Fig. 1(c). Such blank tiles do not add bulk logical legs to a network, but allow us to con-

TABLE II. The ordered set of stabilizers and logical operators that define the block-perfect surface code fragment tensor, $T_{\text{SCF}}^{j_1,\ldots,j_L,j_5} = \langle j_1,\ldots,j_L,j_5|T_{\text{SCF}}\rangle$, where $|T_{\text{SCF}}\rangle$ is the $+1$ eigenstate of S_j's.

Index label:	1	2	3	4	L	5	
	X	X	I	X	$\mathbb{I}$	$\mathbb{I}$	$\equiv S_1$
	$\mathbb{I}$	$\mathbb{I}$	X	X	$\mathbb{I}$	X	$\equiv S_2$
	Z	$\mathbb{I}$	Z	Z	$\mathbb{I}$	$\mathbb{I}$	$\equiv S_3$
	$\mathbb{I}$	Z	$\mathbb{I}$	Z	$\mathbb{I}$	Z	$\equiv S_4$
	X	$\mathbb{I}$	X	$\mathbb{I}$	X	$\mathbb{I}$	$\equiv S_{\bar{X}}$
	$\mathbb{I}$	$\mathbb{I}$	Z	$\mathbb{I}$	Z	Z	$\equiv S_{\bar{Z}}$

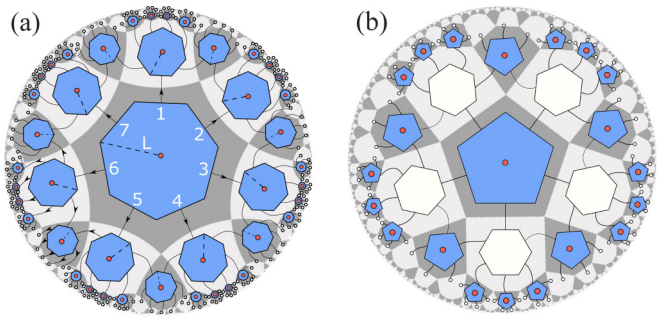

FIG. 2. Graphical tensor network representation of (a) the heptagon code [13] and (b) a reduced-rate pentagon/hexagon code. These describe maps from the red logical qubits in the bulk to the white boundary physical qubits. The seed tensor polygons in (b) could represent any [5, 1] and [6, 0] seed code tensors such as the SCF or five-qubit code.

struct *reduced-rate* codes by mixing such tiles amongst others that do have associated bulk indices. A particular realization of this is the pentagon/hexagon code [3] shown in Fig. 2(b), where blank tiles are interspersed in the tessellation. We note that a given tiling built of order-$(n+k)$ tensors is consistent with any $[n, k]$ code. For instance, the order-6 tensors shown in Fig. 2(b) could represent either the [5, 1, 2] surface code fragment from Fig. 1(a) or the [5, 1, 3] five-qubit code [10] as introduced by Pastawski *et al.* [3].

Lastly, we will also consider tilings in which only a single seed tensor, at the center of the tiling, has a logical qubit associated to it, resulting in a single-logical qubit code with *zero rate*, [3].

IV. PAULI DECODER

In contrast to erasure decoders, optimally correcting Pauli errors for general stabilizer codes is #P-complete [19]. Here we use general purpose integer optimization software to perform most likely error (MLE) correction for general stabilizer codes, albeit with a high computational cost. This is useful for evaluating new QECCs, even if it is impractical for real-time experimental implementations. The algorithm can be applied to any stabilizer code, with suitable adjustments from the CSS variant described here, which we use to compute numerical thresholds for a Pauli error channel for the heptagon code, the SCF code, and the HaPPY code [3].

We represent operators as binary symplectic vectors (BSVs) [30,32,33] for X and Z components defined for n-qubit operator $\hat{A}$ as

$$\hat{A} = \otimes_j \hat{X}^{(\underline{a}_X)_j} \hat{Z}^{(\underline{a}_Z)_j} \equiv \hat{X}^{\otimes \underline{a}_X} \hat{Z}^{\otimes \underline{a}_Z}, \tag{1}$$

where $\underline{a}_X, \underline{a}_Z \in \mathbb{Z}_2^n$. For CSS codes, either $\underline{a}_X$ or $\underline{a}_Z$ will be zero for a given stabilizer or logical operator. Further, for a self-dual CSS code such as the heptagon code, for any X type stabilzser with $(\underline{a}_X, \underline{a}_Z) = (u, 0)$ there is a corresponding Z type stabilizer with $(\underline{a}_X, \underline{a}_Z) = (0, u)$.

To perform decoding we assume X and Z type errors are independent and identically distributed (IID), similarly to the minimum weight perfect matching decoder for surface codes. However, for our simulations we simulate a depolarizing error channel, while decoding with the previous assumption. For clarity we describe a decoder for a self-dual $[n, k]$ CSS QECC subject to Z errors; X errors are treated in the same way. We consider a generic dephasing error given by $\hat{\mathcal{E}} = \hat{Z}^{\otimes \underline{\varepsilon}}$. We define the parity check matrix $\underline{\underline{S}}$, where each stabilizer BSV, $\underline{s}_j$, forms a row of the matrix, so that $\underline{\underline{S}}$ is an $\frac{n-k}{2} \times n$ dimensional matrix. The error syndrome is then given by $\underline{y} = \underline{\underline{S}} \cdot \underline{\varepsilon}$.

From the syndrome we employ an inverse syndrome former (ISF) to find a correction that returns us to the code space. The ISF matrix $\underline{\underline{F}}$ is the pseudoinverse (mod 2) of the parity check matrix, satisfying $\underline{\underline{F}}^T \cdot \underline{\underline{S}}^T = \mathbb{I}$. That is, each column of the ISF defines an operator that anticommutes with only the corresponding stabilizer. It is not always possible to find an ISF in this way; for example in the toric code each error anticommutes with two stabilizers. We use $\underline{\underline{F}}$ along with the syndrome to find a *pure error* BSV, $\underline{e} = \underline{\underline{F}}\, \underline{y}$, that satisfies the syndrome, $\underline{y} = \underline{\underline{S}} \cdot \underline{\varepsilon} = \underline{\underline{S}} \cdot \underline{e}$.

The complete set of errors that satisfy the syndrome is generated by the product of all combinations of stabilizers and logical operators with $\hat{E} = \hat{Z}^{\otimes \underline{e}}$. All such errors have BSV of the form

$$\underline{e}' = \underline{e} + \sum\nolimits_{\ell=1}^{\frac{n-k}{2}} \lambda_\ell \underline{s}_\ell + \sum\nolimits_{m=1}^{k} \mu_m \underline{l}_m, \tag{2}$$

where $\lambda_\ell, \mu_m \in \mathbb{Z}_2$ and $\underline{l}_m$ is a logical operator BSV.

For an IID error model, a MLE decoder minimizes the Hamming weight of e' over λ and μ, to find a most likely correction chain $\underline{c} = \arg\min_{\lambda,\mu \in \mathbb{Z}_2} \mathrm{wt}[\underline{e}']$, where wt is the Hamming weight of the vector. Minimizing wt$[\underline{e}']$ over λ and μ is an integer optimization problem. We implement this optimization problem using the Gurobi optimization package [29]. For the examples shown here Gurobi performs global optimization on the problem.

V. PERFORMANCE SIMULATIONS

We analyze the performance of the codes against Pauli errors using Monte Carlo simulation of an IID depolarizing error model, with error rate p. We generate IID patterns of Pauli errors for a fixed number of errors $a = \mathrm{wt}(\underline{\varepsilon})$, from which we compute the syndrome, which we pass to the integer optimizing decoder. To decide if the decoder has been successful, we calculate the net error after decoding, which is the product of the original error and the correction, $\underline{\varepsilon} + \underline{c}$. The decoder is successful if the net error acts trivially on the logical code space.

We iterate over all $0 \leqslant a \leqslant n$ to estimate the recovery probability, $P_{\mathrm{failure}}(a, n)$, and use the binomial formula

$$p_{\mathrm{failure}}(p, n) = \sum_a \binom{n}{a} p^a (1-p)^{n-a} P_{\mathrm{failure}}(a, n), \tag{3}$$

to calculate the recovery rate for different error rates p.

Sampling over many error instances and different error rates allows us to estimate thresholds in the usual way [7].

We run the decoder for a variety of different tilings and codes. These include heptagon tiling based on the Steane

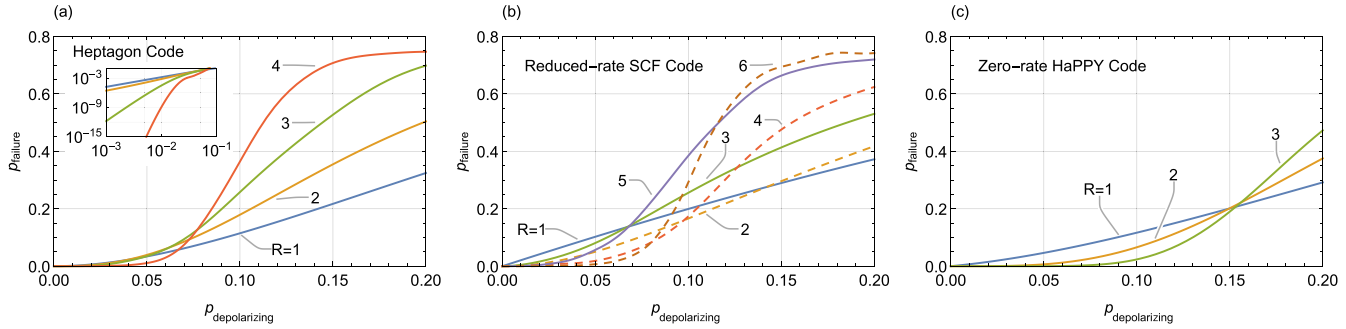

FIG. 3. Failure probability of the central logical qubit in the (a) max-rate heptagon code, (b) reduced-rate SCF code, and (c) zero-rate HaPPY code, in the presence of a depolarizing channel. The inset to (a) shows the same data on logarithmic scale to show the performance well below threshold. Note the error regions around the lines are not significantly wider than the lines themselves. The heptagon code raw data with error bars is shown in Fig. 6 in the Appendix.

QECC, a pentagon tiling code based on the [5, 1, 3] qubit QECC (i.e., the HaPPY code), and a different pentagon tiling based on a [5, 1, 2] surface error-detecting code. For these broad groups, we encode different numbers of logical qubits, ranging from a single logical qubit at the center of the tiling, which asymptotically has zero rate, to maximum-rate encodings with as many logical qubits as possible.

Further, logical qubits in a given code are not homogeneous: logical qubits encoded near the boundary are protected by fewer stabilizers compared to logical qubits encoded closer to the center of the tessellation. As such, here we report threshold figures for the performance of the central logical qubit only.

In Fig. 3(a) we show the failure probability of the central logical qubit for the max-rate heptagon code ($r \approx 0.22$) against IID Pauli errors. With this decoder we see numerical evidence of a threshold near $p \approx 7.0\%$ for the central logical qubit.

This threshold is comparable with the surface code, which has a threshold of 15.5% for the same noise model with a minimum weight perfect matching decoder [23,34], albeit with zero rate.

We also show the performance of the code below threshold in the inset of Fig 3(a), which shows that the logical failure rate, p_{failure}, decreases exponentially with $p_{\text{depolarizing}}$.

In Fig. 3(b) we show the performance of the reduced-rate [5, 1, 2] holographic code. Here we see a pronounced difference between odd radii, where the outer tensors have bulk logical qubits, and even ones. These are characterized by very different codes rates ($r_{\text{odd}} \approx 0.3$ and $r_{\text{even}} \approx 0.09$), and we find thresholds at $p_{\text{odd}} \approx 7.1\%$ and $p_{\text{even}} \approx 8.2\%$ respectively.

Additionally we consider the performance of the zero-rate HaPPY code, shown in Fig 3(c). We see evidence of a threshold here at 16.3%. We note that this was the only non-CSS code we analyzed with the decoder. Even though the code has the same tiling as the SCF (which is CSS), the decoder run time and memory consumption was vastly greater than the decoder applied to CSS codes. As a result, we were only able to estimate the performance of the HaPPY code up to $R = 3$, making this threshold estimate merely provisional.

Figure 4 summarizes a larger set of numerical simulations, plotting the rate and threshold for a variety of different

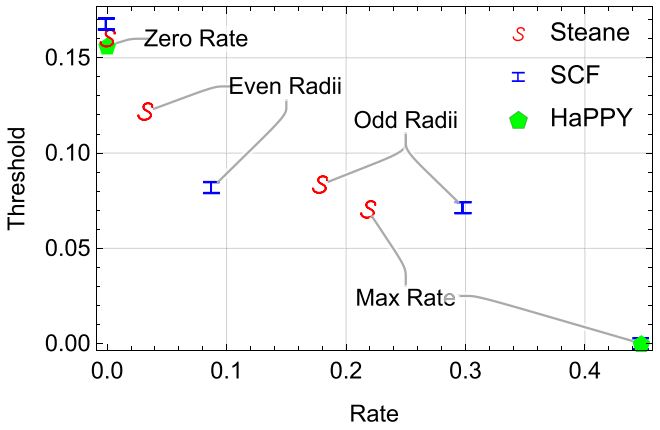

FIG. 4. A comparison of asymptotic rate and numerically estimated threshold against depolarizing errors of holographic codes seeded with the heptagon ("Steane") code, the [5, 1, 3] SCF code, and the five-qubit [5, 1, 3] code (HaPPY). There is an inverse relationship between threshold and rate.

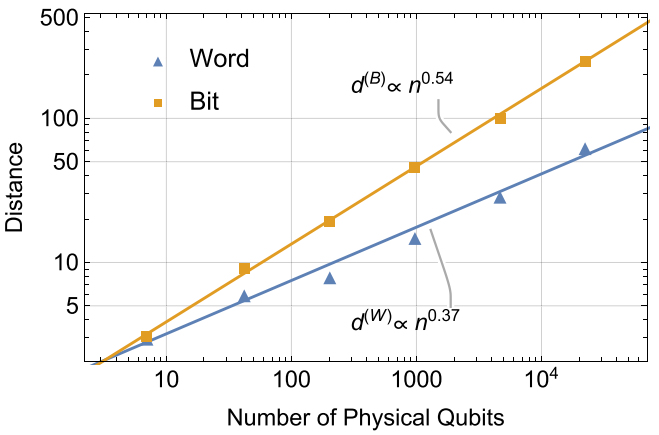

FIG. 5. The distance of the central logical qubit for the heptagon code at different radii; both the bit and word distance scale exponentially with the number of physical qubits. The fit lines are $d^{(W)} = n^{0.37}$ and $d^{(B)} = n^{0.54}$.

TABLE III. Numerically optimized bit and word distance of central logical qubits, with the 95% confidence interval, as a function of radius to the boundary qubits. Additionally we show the asymptotic physical qubit count, n, as a function of the code radius, R, the code rate, r, and extrapolated power-law fits (the numerical prefactor is suppressed) to the numerically computed bit, $d^{(B)}$, and word, $d^{(W)}$, distances of the central logical qubit in each code family.

	Heptagon			Reduced-rate SCF			HaPPY zero rate	
Radius	n	$d^{(B)}$	$d^{(W)}$	n	$d^{(B)}$	$d^{(W)}$	n	$d^{(B)}$
1	7	3	3	5	2	2	5	3
2	42	9	6	25	4	4	25	9
3	203	19	8	75	8	4	95	19
4	973	45	15	255	16	8	355	41
5	4662	99	29	745	20	8	1325	91
6	22337	221	80	2525	40	16	4945	321
Asymptotic	$\sim(5+\sqrt{21})^R$	$\sim n^{0.54\pm0.03}$	$\sim n^{0.37\pm0.07}$	$\sim(5+2\sqrt{6})^R$	$\sim n^{0.31\pm0.10}$	$\sim n^{0.48\pm0.07}$	$\sim(2+\sqrt{3})^R$	$\sim n^{0.65\pm0.08}$

holographic codes. We see the expected tradeoff between rate and threshold quite clearly. Of the codes we have analyzed, none dominates the others: increasing rate never increases the threshold, although there are regions where the tradeoff frontier is notably flat.

VI. DISTANCE

The decoder also provides a way to find the distance of a code, using the integer optimizer to find the minimum weight logical operator. Strictly all maximum rate holographic codes have a fixed distance equal to the distance of the seed code: this is the distance of the logical qubits closest to the boundary. However, bulk qubits further from the boundary are better protected than this distance suggests. A fuller picture is formed by considering the distance of each logical qubit separately.

We define two distance measures for each logical qubit. First the logical *bit distance* is the lowest weight logical operator associated to logical qubit i that acts trivially on all other logical qubits, i.e.,

$$d_i^{(B)} = \min_{\lambda} \mathrm{wt}\Big[\underline{l_i} + \sum\nolimits_{j=1}^{J} \lambda_j \underline{s_j}\Big].$$

Second, the *word distance* is the minimum weight logical operator that has support on logical qubit i, but may act nontrivially on other logical qubits. This distance is equivalent to the distance measure by Pastawski and Preskill [35]. This is calculated using

$$d_i^{(W)} = \min_{\lambda,\mu} \mathrm{wt}\Big[\underline{l_i} + \sum\nolimits_{j=1}^{J} \lambda_j \underline{s_j} + \sum\nolimits_{k\neq i} \mu_k \underline{l_k}\Big].$$

The word distance counts the shortest weight error that would go undetected but would corrupt the data, while the generally larger bit distance counts how large a logical operator needs be to act on an isolated logical qubit, useful, e.g., to work out fault tolerant constructions thereof.

Both of these distances can be found using the same integer optimization package used for decoding. For different code radii in a given code family we compute the code distance, d, and fit a power law as a function of the number of physical qubits. Figure 5 shows the heptagon code distances (for the central logical qubit) against the number of physical qubits. We see numerical evidence of a power-law scaling of distance with the number of physical qubits for both measures. Similar results are found for other code families. Table III shows the results of numerically found code distances for the other codes studied here, along with the results of power-law fits to the numerical results.

We also note that, while the numerics here are for the central logical qubit, the important parameter is the bulk qubit distance from the boundary. We say a bulk qubit is positioned at radius r, which is the number of tensors between the bulk qubit and the center. For example, the radius of the central qubit $r = 0$ and boundary bulk qubits $r = R - 1$. For a given seed tensor and tessellation, the bit distance $d^{(B)}_{R-r}$ is constant for a given value $R - r$; this is also true for word distance.

VII. CONCLUSIONS

We have developed a general purpose decoder for stabilizer codes, based on integer optimization. Using this decoder, we have estimated thresholds against IID depolarizing noise for different families of holographic codes, and have shown that there is a tradeoff between the threshold and the code rate. The moderately high thresholds against phenomenological depolarizing noise are comparable to the surface code. Together with finite rates, they may offer an alternative for building quantum processors. However, the development of an efficient decoder that also has thresholds is essential for implementation. The holographic code stabilizers are nonlocal, so present challenges in implementation; however, generating these codes with low-valence, local cluster states is the subject of ongoing work.

ACKNOWLEDGMENTS

We thank Stephen Wright, Fred Roosta-Khorasani, and David Poulin for useful discussions. This work was supported by the Australian Research Council Centre of Excellence for Engineered Quantum Systems (Grant No. CE 170100009) and

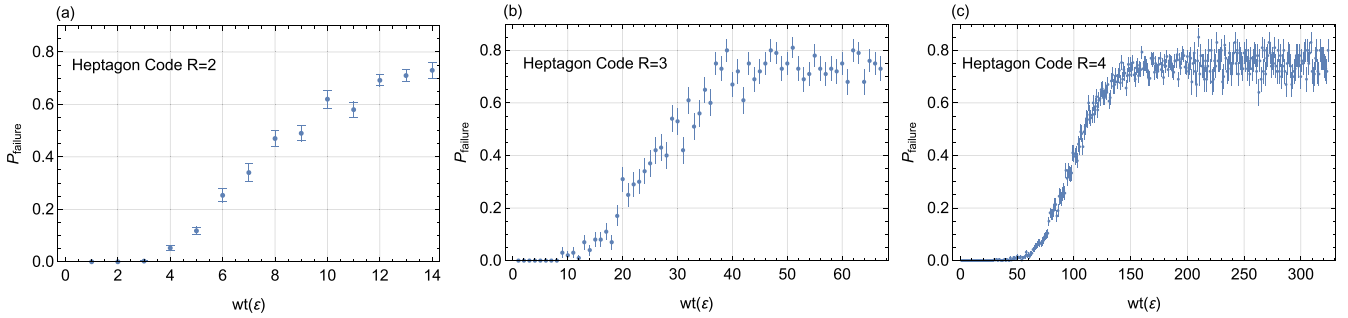

FIG. 6. Failure probability of the central logical qubit, with error bars, in the max-rate Heptagon code for radius (a) $R = 2$, (b) $R = 3$, and (c) $R = 4$, in the presence of a depolarizing channel of fixed weight wt(ε).

the Asian Office of Aerospace Research and Development (AOARD) Grant No. FA2386-18-1-4027.

APPENDIX

We now briefly mention the uncertainties in our analysis. For the failure probability for fixed weight P_{failure}, we estimate the uncertainty in the results as

$$\sigma = \sqrt{\frac{P_{\text{failure}}(1 - P_{\text{failure}})}{m}}, \tag{A1}$$

where m is the number of samples. In Fig. 6 we show the estimate P_{failure} for the max-rate heptagon code with radii 2–4, along with error bars calculated using Eq. (A1).

[1] D. Harlow, Rev. Mod. Phys. **88**, 015002 (2016).
[2] J. I. Latorre and G. Sierra, arXiv:1502.06618.
[3] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill, J. High Energy Phys. 06 (2015) 149.
[4] B. Yoshida, Ann. Phys. (NY) **338**, 134 (2013).
[5] J. Roffe, Contemp. Phys. **60**, 226 (2019).
[6] J. Preskill, Proc. R. Soc. London A **454**, 385 (1998).
[7] S. D. Barrett and T. M. Stace, Phys. Rev. Lett. **105**, 200502 (2010).
[8] T. C. Bohdanowicz, E. Crosson, C. Nirkhe, and H. Yuen, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing - STOC 2019*, Phoenix, June 2019 (ACM, New York, 2019), pp. 481–490.
[9] F. Pastawski and B. Yoshida, Phys. Rev. A **91**, 012305 (2015).
[10] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, Phys. Rev. Lett. **77**, 198 (1996).
[11] T. M. Stace, S. D. Barrett, and A. C. Doherty, Phys. Rev. Lett. **102**, 200501 (2009).
[12] T. M. Stace and S. D. Barrett, Phys. Rev. A **81**, 022317 (2010).
[13] R. J. Harris, N. A. McMahon, G. K. Brennen, and T. M. Stace, Phys. Rev. A **98**, 052301 (2018).
[14] J. Berger and T. J. Osborne, arXiv:1804.03199.
[15] A. Steane, Proc. R. Soc. Math. Phys. Eng. Sci. **452**, 2551 (1996).
[16] J. Preskill, Lecture Notes for Physics 229: Quantum Information and Computation, 1998, http://theory.caltech.edu/~preskill/ph219/index.html (unpublished).
[17] A. Bolt, G. Duclos-Cianci, D. Poulin, and T. M. Stace, Phys. Rev. Lett. **117**, 070501 (2016).
[18] A. Bolt, D. Poulin, and T. M. Stace Phys. Rev. A **98**, 062302 (2016).
[19] P. Iyer and D. Poulin, IEEE Trans. Inf. Theory **61**, 5209 (2015).
[20] M. Sipser and D. A. Spielman, IEEE Trans. Inf. Theory **42**, 1710 (1996).
[21] A. G. Fowler, arXiv:1307.1740.
[22] Y.-H. Liu and D. Poulin, Phys. Rev. Lett. **122**, 200501 (2019).
[23] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[24] N. Delfosse, P. Iyer, and D. Poulin, arXiv:1611.04256.
[25] A. S. Darmawan and D. Poulin, Phys. Rev. E **97**, 051302(R) (2018).
[26] N. Delfosse and N. H. Nickerson, arXiv:1709.06218.
[27] H. Bombin, arXiv:1810.09571.
[28] A. Kubica and N. Delfosse, arXiv:1905.07393.
[29] Gurobi Optimization, LLC, Gurobi optimizer reference manual, 2019, http://www.gurobi.com.
[30] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997 (unpublished), arXiv:quant-ph/9705052.
[31] J. C. Bridgeman and C. T. Chubb, J. Phys. A: Math. Theor. **50**, 223001 (2017).
[32] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, Phys. Rev. Lett. **78**, 405 (1997).
[33] J. Dehaene and B. De Moor, Phys. Rev. A **68**, 042318 (2003).
[34] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, Quantum Inf. Comput. **10**, 456 (2010).
[35] F. Pastawski and J. Preskill, Phys. Rev. X **7**, 021022 (2017).