Random Error Models in Quantum Error Correction

Matthew Coudron and Charles Pasternak

Massachussets Institute of Technology

March 9, 2013

Abstract

We examine the performance of quantum error correcting codes subjected to random Haar distribution transformations of weight t. Rather than requiring correction of all errors, we require some high probability that a random error is corrected. We find that, for any integer i and arbitrarily high probability $p < 1$, there are codes which perfectly correct errors up to weight t and can correct errors up to weight $t + i$ with probability at least p. We also find an analog to the quantum Hamming bound for the new error model. Lastly, we prove that codes generated from classical Reed-Muller codes can correct errors of weight up to $3d/4$ with a probability approaching 1 as the length of the code increases, whereas they can only correct up to weight $d/2$ perfectly.

1 Introduction

Quantum informtion is a relatively young subject, and quantum error correction even more so. Nevertheless, much work has been done in the field. Unlike classical information, quantum information cannot be duplicated [**?**], seemingly rendering classical error correction theory inapplicable. However, Shor proved that quantum error-correcting codes could be constructed [1], and Calderbank and Shor then showed that they can be constructed from classical error-correcting codes [2]. Steane also independently discovered both results [3, 4].

A quantum error-correcting codes is, fundamentally, a method of representing a quantum state composed of k qubits (the quantum analogue of bits) using a quantum state composed of n qubits. This is done by choosing a subspace of the n-qubit state space. These codes are frequently analyzed in terms of the number of qubits in the code, the number of qubits of information encoded, and the severity of the errors they can correct [5, 6].

We look at the last parameter. Past analysis has focused on how severe an error the code can *certainly* correct. We instead ask how severe an error it can *probably* correct. This is based on the observation that real errors are more likely to be random than worst-case; so, instead of wondering about the worst-case error of a certain severity, we worry about the average-case error. We examine how this affects performance of these codes.

2 Background

2.1 Errors

Clearly, no code can protect against all quantum transformations, as such a code could only ever encode a single quantum state. Furthermore, there are an infinite number of possible transformations, and protecting against a finite number is of limited utility. Fortunately, Knill and Laflamme showed that, in order to show that a code corrects all errors in a space E, it is sufficient to show that it corrects all errors in a basis of E [7].

One set of errors that provides an extremely useful basis is the *Pauli group*, which consists of tensor products of the Pauli matrices σ_x, σ_y, σ_z, and σ_i. Pauli group members are their own inverses, and any two members commute or anticommute under composition. In addition, they suggest a natural way of considering the severity of an error. Any Pauli group member may be thought of as independent transformations on each qubit, and some of those transformations are the identity. The *weight* of the member is the number of qubits

which are affected by non-identity transformations. Likewise, more general errors are linear combinations of these Pauli group members, and it makes sense to consider the highest weight of any component of the error.

2.2 Codes

By the quantum no-cloning theorem [**?**], it is impossible to simply duplicate quantum information. However, it can still be spread among multiple qubits [1], which is done by representing a 2^k dimensional space of informational qubits as a subspace C of a 2^n dimensional space of real qubits. The code itself is known as C.

Definition 1. A *quantum error-correcting code* C that represents k qubits in n qubits is a 2^k-dimensional subspace of the 2^n-dimensional state space.

Knill and Laflamme [7] and Gottesman [8] showed that a sufficient condition for C to correct a set of errors E is that the image of C under the error e_i is orthogonal to the image under the error e_j for all $i \neq j$. Equivalently,

$$\langle\psi_1|e_i^\dagger e_j|\psi_2\rangle = \delta_{ij}\langle\psi_1|\psi_2\rangle, \tag{1}$$

where $e_i^\dagger$ is the adjoint of e_i, and δ_{ij} is 1 if $i = j$ and 0 otherwise.

This is not a necessary condition; it states that the code can perfectly distinguish between any two errors, but two errors being indistinguishable does not preclude them being correctable. One example of a situation in which it is false is when $e_i^\perp e_j|\psi\rangle = |\psi\rangle$ for all $|\psi\rangle$. These errors cannot be distinguished, as $e_i|\psi\rangle = e_j|\psi\rangle$. However, because both errors lead to the same result, it does not matter which error occurred: any process to correct the first error would then also correct the second.

Although this condition is not necessary, it is extremely convenient, and it is assumed in the rest of this paper.

2.3 Stabilizer codes

Definition 2. The *stabilizer group* S for a space C is the group of pure transformations T for which $T|\psi\rangle = |\psi\rangle$ for all $|\psi\rangle \in C$. We say C is *stabilized* by S.

Definition 3. The *normalizer group* $S^\perp \supset S$ for a space C is the group of pure transformations which commute with all elements of the stabilizer group for C.

$S^\perp$ is the set of all errors which are undetectable, as they take elements of C to other elements of C. The stabilizer S is the set of all errors which have no effect, as they do not change any elements of C.

Some quantum codes, known as *stabilizer codes*, can be completely specified by their stabilizer groups [8]. One way of representing a stabilizer code is by specifying $n-k$ linearly independent measurements that generate the stabilizer group. All pure errors will either commute or anticommute with each measurement, and the set of measurements an error anticommutes with is known as its *syndrome*. Note that any two errors with different syndromes map C to orthogonal subspaces [8]. Two errors have the same syndrome if and only if their product is in the normalizer group $S^{\perp}$. Let d be the weight of the lowest-weight element of $S^{\perp}$. The weight of the product of two errors is at most the sum of the weights of each error, so no two errors with weight less than $d/2$ can have the same syndrome. Furthermore, if we restrict ourselves to errors that are less than half the weight of any element of S, our code obeys (1) [8].

2.4 Generating Quantum Codes

A common way of generating QECCs is to use classical error correcting codes. Calderbank *et al.* [6] proved that a QECC can be generated by two binary linear classical error correcting codes $C_1 \subseteq C_2$, with length n encoding k_1 and k_2 bits, respectively. The code is stabilized by the group generated as the terms in GF(4) of the form $\omega e_1 + \overline{\omega} e_2$, where $e_1 \in C_1$ and $e_2 \in C_2^{\perp}$. These terms are converted into pure errors using the mapping 0 to I, 1 to Y, ω to X, and $\overline{\omega}$ to Z. The code generated using this method is an $[[n, k_2 - k_1, d]]$ quantum error correcting code, where

$$d = \min\left\{\operatorname{dist}(C_2 \setminus C_1), \operatorname{dist}(C_1^{\perp} \setminus C_2^{\perp})\right\}.$$

3 Probabilistic Error Model

We add a parameter p to the description of codes. Rather than guaranteeing correction for all errors of weight at most t, the code must guarantee that, for any t-site, the probability of correcting a random unitary transformation on that t-site is at least p. In Appendix A, it is proved that the probability of correcting a random error on a site is the proportion of pure errors on the site which are correctable, reducing the calculation of this probability to a counting problem.

This is not equivalent to correcting the proportion p of pure errors of weight at most t. If a code can correct all errors on one t-site but no errors on another, it has an overall probability of 0. This is important because some t-sites may be more likely to have errors than others. For example, qubits that are physically close may be more likely to be affected by the same error than qubits that are far away. To ensure that we

do not overestimate the performance gain from the randomized error model, we take the overall probability of correction to be the lowest probability for any t-site.

4 Specific Cases

To aid in analyzing specific codes, we created a C++ program that finds the probability of correction for a given stabilizer code. The program takes as input the measurements for the code and the value of t to check. It outputs the probability p and a t-site where the probability occurs, and then allows the user to check specific errors. The source code and instructions for use can be found on `http://web.mit.edu/~cpast/Public/check.tar.gz`. The program iterates over all possible errors, noting which syndromes are produced by multiple errors. It then iterates over t-sites, computing the probability for each and returning the lowest overall probability.

Using this method, several improvements to preexisting codes were found. In all cases, only one additional error could be corrected. However, this may be because the codes checked were small codes. As a result of computational limits, we could not use the program to check large codes.

One preexisting code we improved was a $[[21, 5, 6]]$ code, which was found to have $t = 2$ by Calderbank *et al.* [6]. The program confirmed this result, but also indicated that the code can correct errors up to weight 3 with probability $50/64 = 78.125\%$.

5 General Improvements

Further analysis shows that sufficiently large codes can see a fixed improvement with arbitrarily high probability.

Theorem 1. *For any integer m, and for any probability $p_g < 1$, there exists an integer i such that any QECC with a minimum normalizer weight $d > i$ corrects errors up to weight $d/2+m$ with probability greater than p_g.*

Proof. Two errors conflict if and only if their product is in the normalizer group. Because the weight of the product is at most the sum of the weights of the factors, this can only happen if the weights of the errors sum to at least d. Because we only consider errors of weight less than $d/2+m$, any error of weight less than $d/2-m$ cannot be part of a conflict. Because m is fixed, the number of errors with a weight between $d/2-m$ and $d/2+m$ grows much more slowly than the total number of errors with weight less than $d/2+m$. As a

result, when d is sufficiently large, the fraction of errors with a weight high enough to even potentially have a conflict is less than $1 - p_g$.

Note that this analysis does not involve correcting any errors with a weight above $d/2$. Instead, it works because the number of low-weight errors swamps the number of high-weight errors. □

6 Bound

A natural question is to find a bound on the performance of these codes. Ekert and Macchiavello [5] proved the quantum Hamming bound: if a nondegenerate code encodes k qubits in n qubits and can correct all errors of weight at most t, then

$$2^n \geq 2^k \sum_{i=0}^{t} \binom{n}{i} 3^i.$$

A related bound holds for the probabilistic error model. For notational convenience, let us define

$$E_{m,n} = \sum_{i=0}^{n} \binom{m}{i} 3^i$$

as the number of errors of weight at most n on a set of size m.

Theorem 2. *Let C be an $[[n, k, d]]$ stabilizer quantum error correcting code. Let t be a natural number that is less than half of the weight of the lowest-weight element in the stabilizer of C. Let p be the minimum over all t-sites of the probability that a random error on that t-site is correctable. Then*

$$2^n \geq 2^k \sum_{i=0}^{t} \binom{n}{i} 3^i g(t, i, c),$$

where

$$c = \lceil 4^t p \rceil$$

and

$$g(t,i,c) = \begin{cases} 1 & \text{if } c \geq E_{t,i} \\ \dfrac{c - E_{t,i-1}}{\binom{t}{i} 3^i} & \text{if } E_{t,i-1} \leq c \leq E_{t,i} \\ 0 & \text{if } c \leq E_{t,i-1}. \end{cases}$$

Proof. We show that correcting errors on up to t qubits with probability p means that the code must be

able to correct at least

$$\sum_{i=0}^{t} \binom{n}{i} 3^i g(t,i,c)$$

distinct errors. Because t is sufficiently small, the code is nondegenerate. This implies that each error we correct must have a distinct syndrome, and so we can correct only as many errors as there are possible syndromes. As there are $n-k$ measurements in a stabilizer code, and two possible results for each measurement, there are 2^{n-k} possible syndromes in an $[[n,k,d]]$ code. The bound follows immediately.

Rather than randomly selecting an error of weight at most t, we are looking at the probability that we can correct a randomly chosen error at a specific site. The key distinction is that some errors can occur at multiple t-sites. If an error has weight i, then it can occur on any t-site that contains those i qubits. There are $\binom{n-i}{t-i}$ t-sites that meet this requirement.

In order to achieve the required probability, the code will have to correct some set of errors. Consider the smallest set of errors such that the probability that a random error is in the set is at least p. Whether or not an actual stabilizer code exists that corrects exactly this set of errors, any stabilizer code that has the required probability must correct a set of errors at least as large as this set. So, this set gives a lower bound on how many errors must be corrected in order to achieve the required probability.

Errors with low weight occur on more sites than errors with high weight. As a result, the way to achieve the highest probability while correcting the fewest errors is to correct low-weight errors. We want the set of errors to include all low-weight errors before it includes even a single high-weight error. This means that there is some integer k such that an optimal set contains all errors of weight less than k and no errors of weight more than k. These low-weight errors account for most of the probability that we need. To get the rest, we include a portion of the errors of weight k. We have defined the function $g(t,i,c)$ to incorporate this analysis; the derivation of the specifics of the function has been omitted for space. The total number of errors we must correct is

$$\sum_{i=0}^{t} \binom{n}{i} 3^i g(t,i,c). \qquad \square$$

7 Infinite Families

Improvements are not limited to individual codes. At least one entire family of QECCs sees substantial improvement under this probabilistic error model. This family is generated from the Reed-Muller family of classical error correcting codes using the method described in section 2.4. Reed-Muller codes are specified by length and order; we analyzed the codes of length 2^m and order $m/4$. QECCs generated from these codes

can correct $3d/4$ errors with probability approaching 1 as d increases, which is a 50% improvement over standard analysis.

Theorem 3. *Let C_1 be a Reed-Muller code with $r = m/4$, and let C be the code generated from C_1. For this code, $d = 2 \cdot 2^{m/4}$. Under the probabilistic error model, C can correct up to $3d/4$ errors with probability approaching 1 as m increases.*

Proof. We establish a bound on how many errors on a given site are uncorrectable, and show that it grows asymptotically slower than the total number of errors on the site. To do this, we divide the problem into two parts. First, we establish a bound on the number of low-weight normalizer elements of C. Second, we establish a bound on how many uncorrectable errors each normalizer can produce on a single site.

The stabilizer of C is generated as the terms in GF(4) of the form $\omega e_1 + \overline{\omega} e_2$, where $e_1, e_2 \in C_1$. The normalizer is generated as $\omega e_3 + \overline{\omega} e_4$, where $e_1, e_2 \in C_2$. These terms are converted into pure errors using the mapping 0 to I, 1 to Y, ω to X, and $\overline{\omega}$ to Z.

Because we are seeking to correct errors of weight less than $3d/4$, we need not consider normalizer elements with a weight greater than or equal to $3d/2$.

Lemma 3.1. *The number of elements of the normalizer of C which have a weight less than $3d/2$ is $O(2^{3m^2/16+m/4}) = O(n * n^{\log n})$.*

Proof. Berlekamp and Sloane [9] proved that, for Reed-Muller codes with minimum distance d, any codewords with weight between d and $2d$ have a weight which is $2d - 2^i$ for some i. Because d is a power of 2, the largest power of 2 less than d is $d/2$. This means that no codewords can have a weight between d and $2d - d/2 = 3d/2$. Because we can ignore normalizer elements with weight greater than or equal to $3d/2$, we can ignore elements of C_2 with weight greater than d.

A normalizer element generated by elements e_3 and e_4 of C_2 has a weight of $2d$ minus the weight of the intersection of e_3 and e_4. Because Reed-Muller codes are linear, the sum of e_3 and e_4 is a codeword of C_2. This codeword has weight $2d$ minus twice the weight of the intersection of e_3 and e_4. This could be weight zero (if $e_3 = e_4$ or one of them is the identity), or d, or $2d - 2^i$. However, if the weight is $2d - 2^i$, it is at least $3d/2$, so we can ignore it. And if the weight of the codeword is d, then the weight of the normalizer is $3d/2$, so we can ignore this case as well. So, $e_3 = e_4$, or one of e_3 and e_4 is the identity.

We have now reduced our problem of finding low-weight normalizer elements to the problem of finding lowest-weight codewords of C_2. Each such codeword leads to three normalizer elements: one with only X, one with only Y, and one with only Z.

Sloane and Berlekamp [10] proved that the number of lowest-weight elements of a Reed-Muller code $RM(m,r)$ is

$$\frac{2^m}{2^{m-r}} \prod_{i=0}^{m-r-1} \frac{2^{m-i}-1}{2^{m-r-i}-1}. \tag{2}$$

As the normalizer elements are associated with $C_2 = RM(3m/4, m)$, we know that $r = 3m/4$ and (2) reduces to

$$\frac{2^m}{2^{m/4}} \prod_{i=0}^{m/4-1} \frac{2^{m-i}-1}{2^{m/4-i}-1}. \tag{3}$$

The terms of the product can be bounded as

$$\frac{2^{m-i}-1}{2^{m/4-i}-1} < \frac{2^{m-i}}{2^{m/4-i}-1} < \frac{2^{m-i}}{2^{m/4-i-1}} = 2^{3m/4+1},$$

which allows us to rewrite (3) as

$$2^{3m/4}\left(2^{3m/4+1}\right)^{m/4-1} = 2^{3m^2/16+m/4-1}.$$

Because the number of low-weight normalizer elements is three times the number of lowest-weight codewords of C_2, it is

$$O(2^{3m^2/16+m/4-1}) \leq O(2^{m^2+m}),$$

which, since $n = 2^m$, is $O(n * n^{\log n})$. □

The second part of the proof is to bound the number of errors each normalizer can account for.

Lemma 3.2. *If $t \leq 3d/4$, then on any t-site, the number of errors on the t-site which are factors of a normalizer element that intersects the t-site is in*

$$O\left(\frac{3^{5t/3}}{4^{4t/3}}\right).$$

Proof. Say the normalizer element intersects the t-site at i qubits. We seek to find the maximum number of factors.

For each qubit on the t-site, there are two possibilities. Either the error on the qubit is the same as the error on the normalizer, or it is not. If it is the same, e.g. if the factor and the normalizer element both have

an X in the third qubit, then we call it a *correct error*. If it is different, e.g. if the normalizer has an identity error on the fifth qubit but the factor has a Y error there, we call it an *incorrect error*. If a factor has a correct error at a qubit, there is only one possibility for the error at that qubit; if it has an incorrect error, there are three possibilities.

If the intersection of the normalizer and the t-site has size i, there are $d-i$ qubits which are in the normalizer but not the site. For an error to be a factor of the normalizer element, its complement must have correct errors in all $d-i$ of those qubits. Furthermore, the complement must have a non- identity error on each qubit with an incorrect error in the factor. As the complement is of weight at most t, and defining l to be the number of incorrect errors, we have $l \leq t+i-d$. The number of factors associated with t, i, d, and l is

$$\sum_{l=0} t+i-d \binom{t}{l} 3^l.$$

$t+i-d$ grows at most polynomially with t. The terms of the sum grow exponentially. Thus, the sum grows at most as fast as its largest term.

Let $T_l = \binom{t}{l}3^l$ be the term that corresponds to l, and T_{l+1} be that which corresponds to $l+1$. Then

$$\frac{T_{l+1}}{T_l} = \frac{\binom{t}{l+1}3^{l+1}}{\binom{t}{l}3^l} = 3\frac{t-l}{l+1}.$$

This is greater than one when $l < (3t-1)/4$, in which case increasing l will increase T_l. But consider $t+i-d$, which is an upper bound on l. Because $t \leq 3d/4$ and $i \leq t$, this bound must itself be less than or equal to $2t/3$. Thus, $l \leq 2t/3$, which is less than $(3t-1)/4$ as t grows large. This means that $T_l < T_{l+1}$.

Because increasing l always increases the number of factors, the largest term occurs at $l = t+i-d$, and is

$$T_{\max} = \binom{t}{t+i-d} 3^{t+i-d}.$$

We have only one more variable to consider, and that is i. When i increases by one, $T_{\max}$ is multiplied by

$$3\frac{d-i}{t+i-d+1}.$$

But we already analyzed this when we considered $t+i-d$ as an upper bound on l above. We know that the expression is greater than one if t is large. This shows that $T_{\max}$ increases if i increases, so the maximum

is at $i = t$. This gives

$$T_{\max} \leq \binom{t}{2t/3} 3^{2t/3}.$$

Let us work out how fast the binomial coefficient grows. We know that

$$\ln \binom{t}{2t/3} = \ln t! - \ln \frac{t}{3}! - \ln \frac{2t}{3}!. \tag{4}$$

Using Stirling's approximation, we see that (4) is approximately equal to $t(\ln 3 - (\ln 4)/3)$, so the number of factors associated with a codeword grows with

$$\left(\frac{3^{5/3}}{4^{1/3}}\right)^t. \qquad \square$$

The probability an error is uncorrectable thus grows at most as fast as

$$\frac{3^{5t/3} n * n^{\log n}}{4^{t/3} 4^t}.$$

Simplifying, and using the fact that $t = \sqrt[4]{n}$, we have

$$\frac{n^{1+\log n}}{\left(\frac{4^{4/3}}{3^{5/3}}\right)^{\sqrt[4]{n}}}.$$

The base of the bottom exponential term is larger than 1, and $n^{1+\log n}$ grows slower than any exponential, so the number of conflicts grows asymptotically slower than the number of errors, and the probability of correcting $3t/2$ errors approaches one. $\square$

These codes are very sub-optimal under worst-case analysis, so even if they are improved under the probabilistic model, they could still perform worse than some other codes under a worst-case model. However, they are extremely easy to analyze, and these results may translate to other families of codes. Ideally, we would find some asymptotic improvement from a code which performs well under standard analysis. For that, the Reed-Solomon family of codes may show some promise.

8 Conclusion

When properly formalized, the randomization of errors is a physically relevant and mathematically interesting model for analyzing QECCs. Any sufficiently large code can be slightly improved with extremely high probability. This improvement happens even if the code cannot correct any errors that are larger than what it can correct perfectly. Furthermore, there is an infinite family that sees asymptotic improvement, though it is a sub-optimal family of codes. Even the rare failures that do occur under this model can be detected and the computation repeated.

There is still a limit on how well nondegenerate codes can do. The bound from Theorem 2 approaches the quantum Hamming bound as the probability increases. As a result, no families of codes exist that asymptotically exceed the Hamming bound with probability approaching one.

In all of the cases examined, larger codes saw better improvements with higher probabilities, which could mean that the probabilistic model is more useful on states with more qubits. As quantum computers are currently in the experimental stage, and only use a small number of qubits, it may be better to use the worst-case model for now. But if quantum computers are to become practical, they will need more qubits. At that point, it might be useful to use the probabilistic error model.

9 Acknowledgments

I would like to thank the MIT mathematics department, Pavel Etingof, and Tanya Khovanova for organizing and supervising the math projects. Dr. John Rickert and Dr. Jenny Sendova gave invaluable assistance with the paper and presentation, and helped to ensure that the project went smoothly. In addition, I would like to thank Bennett Amodio, Peter Lu, Sitan Chen, Michael Ma, Dominik Rabiej, Scott Kominers, Zachary Abel, Sam Zbarsky, and Ms. Bosse for comments on this paper. Finally, the generous contributions from my sponsors at the Department of Defense and elsewhere allowed me to attend the Research Science Institute, and, without the efforts of the Center for Excellence in Education, that entire program would not have been possible.

References

[1] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52(4):R2493–R2496, Oct. 1995.

[2] A. R. Calderbank and P. W. Shor. Good Quantum Error-Correcting Codes Exist. *Physical Review A*, 54(2):23, Dec. 1995.

[3] A. Steane. Error Correcting Codes in Quantum Theory. *Physical Review Letters*, 77(5):793–797, July 1996.

[4] A. Steane. Multiple-Particle Interference and Quantum Error Correction. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, Nov. 1996.

[5] A. Ekert and C. Macchiavello. Quantum Error Correction for Communication. *Physical Review Letters*, 77(12):2585–2588, Sept. 1996.

[6] A. Calderbank, E. Rains, P. Shor, and N. Sloane. Quantum error correction via codes over GF(4). *IEEE Transactions on Information Theory*, 44(4):1369–1387, July 1998.

[7] E. Knill, R. Laflamme, and L. Viola. Theory of Quantum Error Correction for General Noise. *Physical Review Letters*, 84(11):2525–2528, Mar. 2000.

[8] D. Gottesman. *Stabilizer Codes and Quantum Error Correction.* PhD thesis, California Institute of Technology, May 1997.

[9] E. Berlekamp and N. Sloane. Restrictions on weight distribution of Reed-Muller codes. *Information and Control*, 14(5):442–456, May 1969.

[10] N. Sloane and E. Berlekamp. Weight enumerator for second-order Reed-Muller codes. *IEEE Transactions on Information Theory*, 16(6):745–751, Nov. 1970.

[11] M. Coudron. No Title. Personal communication, July 2012.

A Reduction to Counting Pure Errors

Lemma (Mentor [11])**.** *If U is a $2^n \times 2^n$ complex unitary matrix, chosen at random from a Haar distribution, then the probability that U collapses to a correctable error under measurement is equal to the proportion of pure errors on the n qubits which are correctable.*

Proof. The pure errors G are a basis for unitary matrices. As a result, we can write U as a linear combination

$$U = \sum_{g \in G} c_g g$$

of these errors. Let c_g be the coefficient of $g \in G$ in this combination.

When we make our measurements, we force U to collapse into one of these pure errors. For any g, the probability that U collapses to g is $|c_g|^2$. Because U must collapse to exactly one pure error,

$$\sum_{g \in G} |c_g|^2 = 1.$$

In a Haar distribution, by defintion, every c_g follows the same distribution. This means that

$$E[|c_{g_1}|^2] = E[|c_{g_2}|^2]$$

for all g_1 and g_2 in G. Call this value V.

Call the set of correctable pure errors $G' \subset G$. Then the probability that U collapses to an error in G' is

$$\frac{\sum_{h \in G'} V}{\sum_{g \in G} V}.$$

But this is just

$$\frac{|G'|V}{|G|V} = \frac{|G'|}{|G|},$$

which is the proportion of pure errors that are correctable. □