Pauli error estimation via Population Recovery

Steven T. Flammia[1,2] and Ryan O'Donnell[3]

[1]AWS Center for Quantum Computing, USA
[2]IQIM, California Institute of Technology, USA
[3]Computer Science Department, Carnegie Mellon University, USA

Motivated by estimation of quantum noise models, we study the problem of learning a Pauli channel, or more generally the Pauli error rates of an arbitrary channel. By employing a novel reduction to the "Population Recovery" problem, we give an extremely simple algorithm that learns the Pauli error rates of an n-qubit channel to precision ϵ in ℓ_∞ using just $O(1/\epsilon^2)\log(n/\epsilon)$ applications of the channel. This is optimal up to the logarithmic factors. Our algorithm uses only unentangled state preparation and measurements, and the post-measurement classical runtime is just an $O(1/\epsilon)$ factor larger than the measurement data size. It is also impervious to a limited model of measurement noise where heralded measurement failures occur independently with probability $\leq 1/4$.

We then consider the case where the noise channel is close to the identity, meaning that the no-error outcome occurs with probability $1-\eta$. In the regime of small η we extend our algorithm to achieve *multiplicative* precision $1 \pm \epsilon$ (i.e., additive precision $\epsilon\eta$) using just $O(\frac{1}{\epsilon^2\eta})\log(n/\epsilon)$ applications of the channel.

1 Introduction

A major challenge in the analysis of engineered quantum systems is estimating and modeling noise. The most standard theoretical model for noise in the study of quantum error correction and fault tolerance [26] is the n-qubit *Pauli channel*:

$$\rho \mapsto \sum_{C \in \{0,1,2,3\}^n} p(C) \cdot \sigma_C \rho \sigma_C^\dagger. \tag{1}$$

Here $\sigma_C = \sigma_{C_1} \otimes \cdots \otimes \sigma_{C_n}$ is a tensor product of the Pauli operators $\sigma_0, \sigma_1, \sigma_2, \sigma_3$, and p is a probability distribution on $\{0,1,2,3\}^n$. The numbers $p(C)$ are referred to as the *Pauli error rates*. Additional motivation for the Pauli channel model comes from the practical technique of randomized compiling [18, 29], which converts a general noise channel Λ (with potentially coherent errors) to a Pauli channel Λ_P having the same process fidelity as the original channel. We refer to the $p(C)$ values for Λ_P as the "Pauli error rates" of the original general channel Λ.

Steven T. Flammia: sflammi@amazon.com
Ryan O'Donnell: odonnell@cs.cmu.edu

Given an experimental setup (possibly with randomized compiling), a natural challenge is to diagnose errors in the system via *Pauli error estimation*. Here the goal is to estimate the large Pauli error rates of an unknown channel by preparing states, passing them through the channel, and measuring them. The main desideratum is to minimize the number of measurements; additionally one would like to use simple state preparation and measurement processes and minimal computational overhead. We remark that full tomography for arbitrary n-qubit channels requires at least $4^n/\epsilon^2$ measurements, with more practical methods requiring at least $8^n/\epsilon^2$.

In this work, we give very simple and efficient algorithms for learning all of the large Pauli error rates of an n-qubit channel. Our first main result is the following:

Theorem 1. *There is a learning algorithm that, given parameters $0 < \delta, \epsilon < 1$, as well as access to an n-qubit channel with Pauli error rates p, has the following properties:*

- *It prepares $m = O(1/\epsilon^2) \cdot \log(\frac{n}{\epsilon\delta})$ unentangled n-qubit pure states, where each of the mn 1-qubits states is chosen uniformly at random from $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |i\rangle, |-i\rangle\}$;*
- *It passes these m states through the Pauli channel.*
- *It performs unentangled measurements on the resulting states, with each qubit being measured in either the $\{|0\rangle, |1\rangle\}$-basis, the $\{|+\rangle, |-\rangle\}$-basis, or $\{|i\rangle, |-i\rangle\}$-basis.*
- *It performs an $O(mn/\epsilon)$-time classical post-processing algorithm on the resulting mn measurement outcome bits.*
- *It outputs hypothesis Pauli error rates $\widehat{p}$ in the form of a list of at most $\frac{4}{\epsilon}$ pairs $(C, \widehat{p}(C))$, with all unlisted $\widehat{p}$ values treated as 0.*

The algorithm's hypothesis $\widehat{p}$ will satisfy $\|\widehat{p} - p\|_\infty \leq \epsilon$ except with probability at most δ.

Note that our "sample complexity" of $\widetilde{O}(1/\epsilon^2)$ is optimal up to the logarithmic term: The task of estimating Pauli error rates strictly (and vastly) generalizes the problem of estimating the bias of an unknown coin to additive precision ϵ (and confidence $1 - \delta$), and this is known to require $\Theta(1/\epsilon^2) \cdot \log(1/\delta)$ coin flips. For comparison of our bounds with previous work [10, 14, 15], see §1.2.

When the channel is modeling quantum noise, one hopes and expects that the nontrivial error rate, $\eta = 1 - p(0^n)$, is small. In this case, a natural and more ambitious goal is to first estimate η, and then to estimate all other Pauli error rates to *multiplicative* precision $1 \pm \epsilon$; i.e., additive precision $\pm\epsilon\eta$. (This ambition was also pursued in [10, 14].) Here the ideal sample complexity would be $O(\frac{1}{\epsilon^2\eta})$.[1] If one uses our Theorem 1 as a black box, it would use $\widetilde{O}(\frac{1}{\epsilon^2\eta^2})$ measurements. The extra factor of $1/\eta$ here is quite undesirable (as one might imagine a typical parameter setting to be something like $\eta = 10^{-2}$, $\epsilon = 10^{-1}$). We show that it can be eliminated:

[1] Again, one can compare the task to the vastly simpler one of estimating the face probabilities of a 6-sided die that comes up "1" with probability $1 - \eta$. When rolling many times, one obtains a non-1 outcome roughly every $1/\eta$ rolls. Thus the task becomes very similar to estimating the face probabilities of a 5-sided die to additive precision ϵ, but with a $1/\eta$ "slowdown".

Theorem 2. *In the setting of Theorem 1, suppose the overall error rate is $\eta = 1 - p(0^n)$. One can augment the algorithm so that, given in addition a "noise floor" parameter $0 < \eta_0 < 1$, it has the following properties:*

- *It first makes at most $m_0 := O(1/\eta_0) \cdot \log(1/\delta)$ measurements (as in Theorem 1).*
- *It does $O(m_0 n)$-time classical processing, then either outputs "$\eta \leq \eta_0$" and halts, or proceeds.*
- *It then operates as in Theorem 1, but makes $m := O(\frac{1}{\epsilon^2 \eta}) \cdot \log(\frac{n}{\epsilon\delta})$ measurements.*

Its outputs are correct, with a guarantee of $\|\widehat{p} - p\|_\infty \leq \epsilon\eta$, except with probability at most δ.

Finally, we show that our algorithm can be made impervious to a limited amount of measurement noise. Specifically, suppose that our measuring devices have the following property: When measuring a 1-qubit state from $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |i\rangle, |-i\rangle\}$ in one of the bases $\{|0\rangle, |1\rangle\}$, $\{|+\rangle, |-\rangle\}$, or $\{|i\rangle, |-i\rangle\}$, the device fails (reading out "?") with probability ν, and otherwise behaves ideally. We assume that the failures are independent, and that the algorithm may know the parameter ν (thanks to prior estimation). In this case, we will see that it is almost automatic to obtain the following extension:

Theorem 3. *Theorem 2 continues to hold for any any constant $\nu \leq \frac{1}{4}$.*

For the more challenging task of handling general SPAM (state preparation and measurement) error, see the discussion in §1.2.

1.1 Techniques

Our algorithm employs a novel reduction from Pauli error estimation to the task in classical unsupervised learning known as *Population Recovery*. Population Recovery was introduced by Dvir, Rao, Wigderson, and Yehudayoff in 2012 [9], and has been studied in numerous subsequent works [1–3, 7, 8, 19–21, 23, 25, 30]. A Population Recovery problem is specified by a *classical channel* $\mathcal{S}$ — i.e., a stochastic map $\mathcal{S} : \Sigma \to \Gamma$ for some finite alphabets Σ, Γ. The task is to learn an unknown probability distribution p on Σ^n to ℓ_∞-error ϵ, with the twist being that samples are mediated by the channel. That is, when the learner requests a sample, first $x \in \Sigma^n$ is drawn according to p, but then only $y = \Sigma(x_1)\Sigma(x_2)\cdots\Sigma(x_n)$ is revealed to the learner. The most well-studied cases are the binary symmetric channel and the binary erasure channel, the former being noticeably more challenging; lately, the deletion channel has also begun to be studied. (Each of these channels also requires specifying the crossover/erasure/deletion probability r.)

Our work shows how to efficiently convert the Pauli error estimation task to that of Population Recovery with respect to the so-called *binary Z-channel* with crossover probability $\frac{1}{3}$. This is the channel with $\Sigma = \Gamma = \{0, 1\}$ in which 0's are "transmitted" correctly, but 1's are flipped to 0 with probability $\frac{1}{3}$. We observe that the known methods for Population Recovery with respect to the binary erasure channel with erasure probability r also apply equally well to the Z-channel with crossover probability r. We then use the fact that there is a known, highly efficient Population Recovery algorithm for erasures with probability at most $\frac{1}{2}$. [7, 9, 21, 25] (Indeed, the fact that even probability $\frac{1}{2}$ can be tolerated is the reason our Pauli error estimation algorithm can handle additional measurement noise as in Theorem 3.)

1.2 Previous and related work

There are several prior works that study the estimation of so-called *generalized* Pauli channels, which act on a d-dimensional quantum system and do not contain explicit tensor product structure. These works typically (though not always [27]) make the much stronger assumption that ideal entangled states can be prepared to assist the channel estimation, and they focus on finding efficient estimators that saturate the Cramér-Rao bound. Fujiwara and Imai first showed that entanglement-assisted channel estimation of generalized Pauli channels could achieve the Cramér-Rao bound [12], though much simpler proofs of such theorems are now available [16, Exer. 6.51–6.54]. In particular, these results show that entanglement-assisted estimation of generalized Pauli channels can be done with a sample complexity of $O(1/\epsilon^2)$ in the ℓ_∞ norm.

What about the case of n-qubit Pauli channels in this entanglement-assisted setting? An ϵ-close estimate in the ℓ_∞ norm is also achievable with only $O(1/\epsilon^2)$ samples in this setting. This can be seen by noting that inputting half of a maximally entangled state into a Pauli channel and measuring in a Bell basis gives completely distinguishable outcomes for each Pauli error[2]. The problem therefore reduces to estimating a *classical* probability distribution on 4^n outcomes, and for this well-studied problem the sample complexity is well known to be $\Theta(1/\epsilon^2)$ (see Ref. [5] for a simple proof). In light of this, one way to interpret our Theorem 1 is that entanglement-free estimation of Pauli channels is at most a logarithmic factor away from the optimal sample complexity, at least for the ℓ_∞ norm.

The problem of Pauli error estimation for n-qubit channels without entanglement was first studied in depth in work of the first author and Wallman [10]. It is not possible to directly compare those results with ours, for several reasons. The most immediate reason is that their complexity bounds typically include a factor of $\widetilde{O}(1/\Delta)$, where "Δ" is another parameter, the spectral gap of the Pauli channel being learned. We have $\Delta \leq 2\eta$, where $\eta = 1 - p(0^n)$ is the nontrivial error rate, and this is saturated in the most favorable case. However, in general Δ may be arbitrarily small, or even zero, for relatively simple channels. In practice, a user of the algorithm in [10] would set a spectral cutoff Δ_0 and allow estimation errors for channel eigenvalues in the interval $(1-\Delta_0, 1]$, but no analysis is done in [10] of the extra error incurred by this cutoff. Thus, in the worst case, their results as formally stated do not give any guarantee.

On the other hand, the results of [10] are impervious to a much more challenging model of measurement error ("SPAM"). This model imposes that before the learner measures the channel's output, an *additional* unknown channel Ξ is applied to the state. (It is assumed that Ξ satisfies the extremely mild condition that its nontrivial error rate is bounded away from 1.) It might seem impossible to disentangle Ξ from the main channel Λ to be learned, but the authors of [10] use the fact that one is at liberty to pass a state ρ through Λ several times (say, k times) before it is subjected to Ξ; i.e., the learner may obtain $\Xi\Lambda^k\rho$ for ρ and $k \in \mathbb{N}$ of the learner's choosing. By carefully choosing k values up to $O(1/\Delta)$, the authors of [10] show that Ξ can essentially be expunged. (Note that, in practice, multiple uses of the channel are often far less costly than even a single measurement.)

Finally, the first algorithm in [10] judges its hypothesis with respect to the ℓ_2-norm, rather

[2] This is essentially superdense coding [4]. One can show by computing the diamond norm of the difference between two Pauli channels that this strategy has optimal sample complexity up to a constant factor.

than the ℓ_∞ norm as in this paper. This distinction is relatively minor, however, as the norms are roughly equivalent for probability distributions: $\|\widehat{p} - p\|_\infty \leq \|\widehat{p} - p\|_2 \leq \|\widehat{p} - p\|_\infty^{1/2}$, and one may refine this further to take into account dependence on $\eta = 1 - p(0^n)$.

With these caveats, we state (simplifications of) the relevant main results in [10]:

Theorem 4 ([10])**.** *There exists a SPAM-tolerant algorithm that makes $\widetilde{O}(2^n \log(1/\Delta))/\epsilon^2$ measurements, with $O(1/\Delta)$ channel-uses per measurement, and with high probability outputs an estimate $\widehat{p}$ of the channel's Pauli error rates p satisfying $\|\widehat{p} - p\|_2 \leq \epsilon\eta$.*

In the favorable case of $\Delta = \Theta(\eta)$, this is somewhat comparable to our Theorem 2; the above theorem has much better SPAM-tolerance, but a complexity that is greater by roughly 2^n.

The authors of [10] also present a heuristic for identifying a set S corresponding to large Pauli error rates with the following guarantee.

Theorem 5 ([10])**.** *For any set $S \subseteq \{0,1,2,3\}^n$, there exists a SPAM-tolerant algorithm that makes $\widetilde{O}(\log|S|)\log\log(1/\Delta)/\epsilon^4$ measurements, with $O(1/\Delta)$ channel-uses per measurement, and with high probability outputs estimates $\widehat{p}(C)$ for each $C \in S$ satisfying $|\widehat{p}(C) - p(C)| \leq \epsilon\eta$.*

However, no guarantee is proven that the set S will contain the $|S|$ largest error rates.

The results in [15] are also somewhat incomparable to the present paper. The authors analyze Pauli channels with a recovery guarantee in the ∞-norm, but under the assumption that the Pauli channel has sparse and random support, and that the nonzero error rates are not too small (greater than some fixed ϵ_0). While the sparsity assumption is not critical in that analysis (the algorithm will approximate error rates smaller than ϵ_0 as zero with high probability), the random support assumption is used in an essential way. This is an undesirable assumption since it is very unlikely to hold in practice.[3] The sample complexity is also not stated directly in terms of quantum measurements, but rather in terms of queries to a "noisy eigenvalue oracle" with Gaussian noise. While this noisy oracle can be approximated by quantum measurements and finite sample complexity, quantum noise is not exactly Gaussian, so no direct comparison with the present work is possible without further analysis.

We remark that the techniques used in [10, 15] are Fourier-based, and the heuristic from [10] described above is similar to the Goldreich–Levin learning algorithm [13]. In §7, we give an alternate Fourier-based approach to Pauli error estimation, one that is equivalent to our Population Recovery method "in disguise"; in fact, the Goldreich–Levin algorithm becomes equivalent to the Individual-to-Population Recovery reduction!

It is our belief that these Fourier techniques can actually be used to provide a common generalization of the results of this paper and of [10]; i.e., efficient SPAM-tolerant Pauli error estimation with no dependence on Δ. We leave this for future work.

2 Notation

Notation 6. The 1-*qubit Pauli matrices* are the unitary, hermitian matrices

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \sigma_1 = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_2 = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_3 = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

[3] Perhaps surprisingly, the algorithm performs well on real data despite grossly violating this assumption [15].

As operators on the Bloch sphere, $\sigma_1, \sigma_2, \sigma_3$ act as rotations by π about the 1st, 2nd, 3rd axis (aka x-, y-, z-axis), respectively. More generally, an *n-qubit Pauli matrix*, indexed by string $A \in \{0,1,2,3\}^n$, is $\sigma_A = \bigotimes_{j=1}^n \sigma_{A_j}$.

Notation 7. For $a, b \in \{0,1,2,3\}$, there is some $c \in \{0,1,2,3\}$ such that $\sigma_a \sigma_b = \sigma_c$, up to a global phase. We introduce the notation $a \oplus b$ (equivalently, $b \oplus a$) for this c; so, e.g, $1 \oplus 3 = 2$, $0 \oplus b = b$, etc. We extend the notation coordinate-wise: if $A, B \in \{0,1,2,3\}^n$, then $A \oplus B = (A_1 \oplus B_1, \ldots, A_n \oplus B_n) \in \{0,1,2,3\}^n$ (and so $\sigma_A \sigma_B = \sigma_{A \oplus B}$, up to a global phase).

Notation 8. We write the orthonormal eigenbasis for the Pauli operator σ_x as $|\chi^1_+\rangle, |\chi^1_-\rangle$. On the Bloch sphere these are the two unit vectors pointing in the positive (respectively, negative) direction along the 1st (x-)axis; they are often called $|+\rangle$, $|-\rangle$. We use similar notation $|\chi^2_+\rangle, |\chi^2_-\rangle$ (often called $|i\rangle, |-i\rangle$) and $|\chi^3_+\rangle, |\chi^3_-\rangle$ (often called $|0\rangle, |1\rangle$) for σ_2 and σ_3.

Notation 9. For $a, b \in \{0,1,2,3\}$ we have that $\sigma_b |\chi^a_+\rangle$ is (up to a phase) $|\chi^a_\pm\rangle$, with the subscript being $+$ if σ_a and σ_b commute, and $-$ if σ_a and σ_b anticommute. To capture this, it will be convenient to introduce the following notation:

$$a \star b = b \star a = \begin{cases} \mathtt{0} & \text{if } |\{a, b, a \oplus b\}| < 3, \text{ i.e., } \sigma_a, \sigma_b \text{ commute;} \\ \mathtt{1} & \text{if } |\{a, b, a \oplus b\}| = 3, \text{ i.e., } \sigma_a, \sigma_b \text{ anticommute.} \end{cases}$$

Thus $\sigma_b |\chi^a_+\rangle = |\chi^a_{(-1)^{a \star b}}\rangle$ (up to a phase). We extend this notation coordinate-wise, writing $A \star B = (A_1 \star B_1, \ldots, A_n \star B_n) \in \{\mathtt{0}, \mathtt{1}\}^n$ for $A, B \in \{0,1,2,3\}^n$. For example, $(0,0,3,2,1) \star (3,1,1,2,2) = (\mathtt{0}, \mathtt{0}, \mathtt{1}, \mathtt{0}, \mathtt{1})$.

Fact 10. If we identify $\{0,1,2,3\}$ with $\mathbb{F}_2^2$ by writing numbers in base 2, then $\oplus$ corresponds to the usual vector addition in $\mathbb{F}_2^2$, and $\star$ corresponds to the "symplectic" product: $a \star b = (a_1, a_2) \star (b_1, b_2) = a_1 b_2 + a_2 b_1$. This lets us see that $a \star (b \oplus c) = (a \star b) + (a \star c) \bmod 2$.

Notation 11. For a quantity x, we denote an estimate of x by $\widehat{x}$. We use boldface font (e.g., $\boldsymbol{A}$) to denote a random variable. If $\boldsymbol{A}$ is drawn from the distribution p we denote this by $\boldsymbol{A} \sim p$, and let A denote a concrete assignment to the variable $\boldsymbol{A}$. Addition (of scalars or vectors) modulo 2 is denoted $+_2$. The Fourier transform of f is denoted $\tilde{f}$.

3 Learning a Pauli channel

In this section we describe the basic setup for learning a Pauli channel. Learning the Pauli error rates of a general channel will end up being just a minor extension, discussed in §6.1.

As described in Equation (1), an n-qubit Pauli channel is determined by a probability distribution p on $\{0,1,2,3\}^n$. This probability distribution induces the mixed unitary channel in which σ_C is applied with probability $p(C)$. An $n = 5$ example:

$$p(00321) = 2/10, \quad p(01300) = 3/10, \quad p(11323) = 2/6, \quad p(30000) = 1/6, \quad p(C) = 0 \text{ otherwise.}$$

We anthropomorphize by imagining a character Charlie who operates the channel; on receiving a state ρ, Charlie first (secretly) draws $\boldsymbol{C} \sim p$, then outputs the state $\sigma_{\boldsymbol{C}} \rho$.

Alice the Learner would like to estimate the probability distribution p via interactions with Charlie. Alice has the ability to prepare n-qubit states, to "query" Charlie (i.e., pass an n-bit

state through his channel), and to measure states that she receives back. Her goal is to learn a precise approximation to p (with high probability), while minimizing the number of queries to Charlie.

Definition 12. We say that Alice performs a *nontrivial probe* if she does the following:

- She chooses a string $A \in \{1,2,3\}^n$.
- She prepares the (unentangled) n-qubit state $|\psi_A\rangle$ in which the jth qubit is $|\chi_+^{A_j}\rangle$.
- She passes $|\psi_A\rangle$ through Charlie, obtaining $\sigma_C |\psi_A\rangle$ with probability $p(C)$.
- She does a (non-entangled) measurement on the resulting n-qubit state, measuring the jth qubit in the basis $|\chi_\pm^{A_j}\rangle$.

Continuing our $n = 5$ example, if Alice does a nontrivial probe with the string $A = 31122$, this entails preparing and passing to Charlie the state

$$|\psi_{31123}\rangle = |\chi_+^3\rangle\,|\chi_+^1\rangle\,|\chi_+^1\rangle\,|\chi_+^2\rangle\,|\chi_+^2\rangle \quad \Big(= |0\rangle\,|+\rangle\,|+\rangle\,|i\rangle\,|i\rangle\Big),$$

and then measuring the 5 returned qubits in the bases $|\chi_\pm^3\rangle$, $|\chi_\pm^1\rangle$, $|\chi_\pm^1\rangle$, $|\chi_\pm^2\rangle$, $|\chi_\pm^2\rangle$, respectively.

Now suppose that Charlie drew $C = 00321$ (which occurs with probability 2/10 in our example). Then the state returned to Alice would be

$$\begin{aligned}(\sigma_0 \otimes \sigma_0 \otimes \sigma_3 \otimes \sigma_2 \otimes \sigma_1)\,|\psi_{31122}\rangle &= (\sigma_0\,|\chi_+^3\rangle) \otimes (\sigma_0\,|\chi_+^1\rangle) \otimes (\sigma_3\,|\chi_+^1\rangle) \otimes (\sigma_2\,|\chi_+^2\rangle) \otimes (\sigma_1\,|\chi_+^2\rangle) \\ &= e^{i\theta} \cdot |\chi_+^3\rangle\,|\chi_+^1\rangle\,|\chi_-^1\rangle\,|\chi_+^2\rangle\,|\chi_-^2\rangle\end{aligned}$$

for some phase $e^{i\theta}$ ($\theta \in \mathbb{R}$) that we did not bother to compute. Now when Alice measures in the bases $|\chi_\pm^3\rangle$, $|\chi_\pm^1\rangle$, $|\chi_\pm^1\rangle$, $|\chi_\pm^2\rangle$, $|\chi_\pm^2\rangle$, her readout will, with probability 1, be

$$|\chi_+^3\rangle\,|\chi_+^1\rangle\,|\chi_-^1\rangle\,|\chi_+^2\rangle\,|\chi_-^2\rangle\,.$$

The subscripts $+, +, -, +, -$ here are the 5 bits of information conveyed to Alice by the readout, and we may think of instead labeling them as `00101` in accordance with Notation 9. With this relabeling convention, we obtain:

Fact 13. Suppose Alice performs a nontrivial probe with string $A \in \{1,2,3\}^n$, and suppose the random string drawn by Charlie is $C \in \{0,1,2,3\}^n$. Then when Alice measures, she obtains the readout $R = A \star C \in \{\mathtt{0},\mathtt{1}\}^n$.

Remark 14. So far we have pictured Alice as first choosing A, and then Charlie as drawing a random C. It is useful now to make a slight shift in perspective: for each interaction between Alice and Charlie, we will equivalently think of *Charlie* as first (secretly) drawing C, and then Alice gaining some partial information about this C by "probing" it using an A of her choice. We emphasize that Alice must make her choice of A without knowing the channel outcome C.

We now describe a trick that Alice may employ in probing the channel:

Definition 15. For a channel distribution p on $\{0,1,2,3\}^n$, and any fixed $B \in \{0,1,2,3\}^n$, define the *B-altered channel distribution* $p^{\oplus B}$ on $\{0,1,2,3\}^n$ via $p^{\oplus B}(C) = p(B \oplus C)$.

For any string $B \in \{0,1,2,3\}^n$ of her choosing, Alice can effectively simulate access to the B-altered channel: If she wishes to simulate passing $|\phi\rangle$ through the B-altered channel, she could instead simply pass $\sigma_B |\phi\rangle$ through Charlie's actual channel. (This may introduce a "wrong" global phase, but it doesn't matter for any measurement behavior that we consider here.) But in fact, something even simpler is true:

Observation 16. Given $B \in \{0,1,2,3\}^n$, if Alice wants to perform a nontrivial probe of $p^{\oplus B}$ based on string A, she can pass $|\psi_A\rangle$ to Charlie as always. Then, when she measures and obtains $A \star C$, she can "reinterpret" this readout by adding in, mod 2, the string $A \star B \in \{\mathtt{0},\mathtt{1}\}^n$ (which she knows). Recalling Fact 10, this gives her $(A \star B) +_2 (A \star C) = A \star (B \oplus C)$. Thus the reinterpreted readout is indeed distributed as what she would get by probing $p^{\oplus B}$ with A.

A natural strategy for Alice is to make *random* nontrivial probes. It is easy to see the following:

Fact 17. Fix a draw $C \in \{0,1,2,3\}^n$ for Charlie. Now if Alice performs a nontrivial probe with a uniformly random $\boldsymbol{A} \in \{1,2,3\}^n$, then the coordinates of her readout $\boldsymbol{R} = \boldsymbol{A} \star C \in \{\mathtt{0},\mathtt{1}\}^n$ will be independent, with the following distribution for each $1 \le j \le n$:

- If $C_j = 0$ then $\boldsymbol{R}_j$ will be $\mathtt{0}$ with probability 1.
- If $C_j \neq 0$ then $\boldsymbol{R}_j$ will be $\mathtt{0}$ with probability $\frac{1}{3}$ and $\mathtt{1}$ with probability $\frac{2}{3}$.

We can state this more succinctly by introducing some additional terminology:

Notation 18. For $B, C \in \{0,1,2,3\}^n$, define the string $C^{\neq B} \in \{\mathtt{0},\mathtt{1}\}^n$ by

$$(C^{\neq B})_j = \begin{cases} \mathtt{1} & \text{if } C_j \neq B_j, \\ \mathtt{0} & \text{if } C_j = B_j. \end{cases}$$

Definition 19. Recall from information theory the so-called *Z-channel with crossover probability r*: it is the binary channel that leaves $\mathtt{0}$ untouched and flips $\mathtt{1}$ to $\mathtt{0}$ with probability r.

Now Fact 17 can be restated as follows:

Fact 20. Fix a draw $C \in \{0,1,2,3\}^n$ for Charlie. Now if Alice performs a random nontrivial probe, her readout is the result of passing $C^{\neq 0^n}$ through a Z-channel with crossover probability $\frac{1}{3}$.

Observation 21. By combining Observation 16 with Fact 20, we obtain the following: Fix a draw $C \in \{0,1,2,3\}^n$ for Charlie and suppose Alice performs a random nontrivial probe. She can then — for any fixed $B \in \{0,1,2,3\}^n$ — interpret her readout as $C^{\neq B}$ passed through a Z-channel with crossover probability $\frac{1}{3}$. *Warning:* these reinterpretations are completely *dependent*; she of course cannot get the result of *independent* channel applications for various B's, unless she makes multiple probes.

4 Population Recovery

With Observation 21 in hand, we have effectively reduced the problem of learning a Pauli channel to a "Population Recovery"-type problem (with a quantum-free definition). To recap: there is an unknown probability distribution p on $\{0,1,2,3\}^n$, a learner may request samples, and when a sample C is drawn from p, the learner receives a binary string which can be interpreted as "$C^{\neq B}$ passed through a Z-channel with crossover $\frac{1}{3}$" for any $B \in \{0,1,2,3\}^n$ of the learner's choosing.

In this section we will give a solution to this problem that has optimal sample complexity (except possibly up to a logarithmic factor) using techniques from the field of Population Recovery. Our solution will immediately imply Theorem 1 in the special case where the channel to be learned is indeed a Pauli channel. The case of learning a *general* channel's Pauli error rates is treated in §6.1. We remark that our Pauli channel algorithm only uses nontrivial probes, and thus only involves preparing the states $|0\rangle$, $|+\rangle$, and $|i\rangle$. The other three states $|1\rangle$, $|-\rangle$, and $|-i\rangle$ are only used for the extension to general channels.

Idea of our solution. Using known techniques from Population Recovery, one can first reduce to the simpler task of "Individual Recovery" (estimating a single $p(B)$ value) via a coordinate-by-coordinate learning algorithm. Then one can further reduce to just recovering $p(0^n)$, using the altered-channel trick. As for learning $p(0^n)$, we first observe that the replacement of C by $C^{\neq 0^n}$ changes nothing for this problem, so we effectively have the same task just for the $\frac{1}{3}$-crossover Z-channel on binary strings. This is similar to the erasure channel with erasure probability $\frac{1}{3}$, and in fact the known solutions for erasure probability-r [7, 9, 21, 25] *only use the locations of the* 1*'s in the received word.* Thus these known solutions work *equally well* for the Z-channel. Indeed, as noted in [9], the solution is particularly simple when $r \leq \frac{1}{2}$ (as it is for us); the full method of "robust local inverses" is not needed, and one can use the "natural inverse" (as we implicitly do in the proof of Theorem 22 below).

4.1 Individual Recovery

Although the proof of the below theorem is self-contained, we remark that it implicitly follows the Individual Recovery routine of [9] for the $\frac{1}{3}$-erasure channel.

Theorem 22. *For any fixed $B \in \{0,1,2,3\}^n$, a version of Theorem 1 holds in which the learner only computes an estimate $\widehat{p}(B)$ of $p(B)$ satisfying $|\widehat{p}(B) - p(B)| \leq \epsilon_0$ except with probability at most δ_0. The number of samples used is $m = O(1/\epsilon_0^2) \cdot \log(1/\delta_0)$ and the classical post-processing time is $O(mn)$.*

Remark 23. The reader may wish to verify the proof just in the case $B = 0^n$, where it is simpler; the general case then follows from Observation 16.

Proof. Alice obtains m probe/readout pairs $(\boldsymbol{A}, \boldsymbol{R})$, with $\boldsymbol{A} \sim \{1,2,3\}^n$ uniformly random and $\boldsymbol{R} = \boldsymbol{A} \star \boldsymbol{C}$, where $\boldsymbol{C}$ is a random channel outcome drawn from p. The estimate $\widehat{p}(B)$ that Alice will output is the empirical mean of the random variable

$$\boldsymbol{H} = (-1/2)^{|\boldsymbol{A} \star B +_2 \boldsymbol{R}|} = (-1/2)^{\sum_t ((\boldsymbol{A} \star B) +_2 \boldsymbol{R})_t} = \prod_{t=1}^{n} (-1/2)^{\boldsymbol{y}_t}, \quad \boldsymbol{y}_t \coloneqq (\boldsymbol{A}_t \star B_t) +_2 \boldsymbol{R}_t.$$

As seen in Observation 21, for a given outcome $\boldsymbol{C} = C$, the random binary string $(\boldsymbol{A} \star B) +_2 \boldsymbol{R}$ is distributed as $C^{\neq B}$ passed through a Z-channel with crossover probability $\frac{1}{3}$. In particular, its coordinates $\boldsymbol{y}_t$ are independent random variables, with conditional expectation given by

$$\mathbf{E}[(-1/2)^{\boldsymbol{y}_t} \mid \boldsymbol{C} = C] = \begin{cases} (-1/2)^0 = 1 & \text{if } C_t = B_t, \\ \frac{1}{3}(-1/2)^0 + \frac{2}{3}(-1/2)^1 = 0 & \text{if } C_t \neq B_t. \end{cases}$$

Thus

$$\mathbf{E}[\boldsymbol{H} \mid \boldsymbol{C} = C] = \prod_{t=1}^{n} \mathbf{E}[(-1/2)^{\boldsymbol{y}_t} \mid \boldsymbol{C} = C] = \begin{cases} 1 & \text{if } C = B, \\ 0 & \text{if } C \neq B, \end{cases}$$

and hence indeed $\mathbf{E}[\boldsymbol{H}] = p(B)$. □

4.2 Population Recovery

Theorem 22 allows Alice to estimate $p(B)$ for any particular string $B \in \{0,1,2,3\}^n$. But also, for any shorter string $\beta \in \{0,1,2,3\}^\ell$, Alice can estimate the marginal

$$p(\beta) \coloneqq \sum_{\gamma \in \{0,1,2,3\}^{n-\ell}} p(\beta\gamma) = \mathop{\mathbf{Pr}}_{C \sim p}[(C_1, \ldots, C_\ell) = \beta],$$

simply by ignoring all data in positions $\ell+1, \ldots, n$. (She is obviously not limited to marginalizing contiguous blocks, but this is all we will need for our purposes.) Alice can thus learn all of p to good ℓ_∞-precision with the straightforward, coordinate-by-coordinate branch-and-prune approach common in Population Recovery (see, e.g., [25, App. A]). We repeat this approach here; the following algorithm achieves our main Theorem 1 for Pauli channels, except for the claim about the running time of the post-processing algorithm:

1. Set $\epsilon_0 = \frac{\epsilon}{4}$, $\delta_0 = \frac{4\epsilon\delta}{9n}$ and draw a single batch of m samples, where m is as in Theorem 22.
2. Define "support sets" $\Omega_1 = \{0,1,2,3\}$ and $\Omega_2 = \cdots = \Omega_n = \emptyset$.
3. For round $j = 1 \ldots n-1$:
4. For each prefix $\beta' \in \Omega_j$ and each $b \in \{0,1,2,3\}$:
5. Run the Individual Recovery algorithm on $\beta \coloneqq \beta' b$ to estimate the marginal $p(\beta)$.
6. If the estimate is at least $2\epsilon_0 = \frac{\epsilon}{2}$, then place β into Ω_{j+1}.
7. Output as $\widehat{p}$ the collection of strings in Ω_n, together with their estimated probabilities.

The correctness of the algorithm, that $\|\widehat{p} - p\|_\infty \leq \epsilon$ with failure probability at most δ, is straightforward and is explicitly proven in [25, Lem. 18]. The proof also establishes that when there is no failure, $|\Omega_j| \leq \frac{4}{\epsilon}$ holds for all $1 \leq j \leq n$. Thus for running time purposes (and without impacting the correctness claim) we may have the algorithm abort if ever some Ω_j gets cardinality more than $\frac{4}{\epsilon}$. It only remains to obtain the post-processing running time of $O(mn/\epsilon)$ claimed in Theorem 1.

Running time analysis. As it stands, the running time of the above algorithm is $O(mn^2/\epsilon)$, since it may do up to $O(n/\epsilon)$ executions of the $O(mn)$-time Individual Recovery algorithm. (We have implemented this naïve version of the algorithm in Julia [11] for interested readers.) However, since all executions of the Individual Recovery algorithm are on the same batch of samples, it's not hard to see that information from the jth round of the algorithm can be used to speed up the $(j+1)$st round. More precisely, we show that each round can be done in $O(m/\epsilon)$ time, leading to the overall claimed running time of $O(mn/\epsilon)$.

Let $R \in \{0,1\}^{m \times n}$ be the measurement outcome bits that the algorithm processes, and let $R_{1\ldots j}$ denote the submatrix formed by the first j columns. Also, for $\beta \in \{0,1,2,3\}^j$, let $R^{(\beta)} \in \{0,1\}^{m \times j}$ be the (hypothetical) matrix whose tth row is the same as $R_{1\ldots j}$'s but with $(A_1^t, \ldots, A_j^t) \star \beta$ added in mod 2, where A^t is the tth probe string used by Alice. Given β, the algorithm can look up entries of $R^{(\beta)}$ in $O(1)$ time.

Recall that when the algorithm does Individual Recovery on the prefix β, it computes the fraction of rows of $R^{(\beta)}$ that have Hamming weight i, multiplies this number by $(-1/2)^i$, and sums the results. In particular, this estimate can be computed in $O(m)$ time given the vector $h^{(\beta)} \in \mathbb{N}^m$ whose tth entry is the Hamming weight of the tth row of $R^{(\beta)}$ — just add up $(-1/2)^{h_t^{(\beta)}}/m$ across all t.

We can now modify the above Population Recovery algorithm so that whenever a prefix $\beta \in \{0,1,2,3\}^j$ is added into Ω_j, the algorithm retains the vector $h^{(\beta)}$ that went into estimating $p(\beta)$. It is easy to see that in the subsequent round, we can compute each of $h^{(\beta 0)}, h^{(\beta 1)}, h^{(\beta 2)}, h^{(\beta 3)}$ from $h^{(\beta)}$ (and hence the marginal estimates) in $O(m)$ time, and retain them as needed. Thus indeed each round only requires $O(m/\epsilon)$ time, since at most $\frac{4}{\epsilon}$ prefixes are processed in each round.

5 Multiplicative error

In a practical scenario we would would hope that the "nontrivial error rate" of the Pauli channel,

$$\eta \coloneqq 1 - p(0^n)$$

is very small. This motivates writing p as a mixture distribution, as follows:

$$p: \quad \text{mixing weight } 1-\eta \text{ on } 0^n, \quad \text{mixing weight } \eta \text{ on } p_{\text{err}}, \tag{2}$$

where p_{err} is a distribution on $\{0,1,2,3\}^n \setminus \{0^n\}$. Now a natural goal is to learn with *multiplicative error* ϵ, meaning producing estimates $\widehat{\eta}$, $\widehat{p}_{\text{err}}$ with

$$(1-\epsilon)\eta \le \widehat{\eta} \le (1+\epsilon)\eta, \qquad \|\widehat{p}_{\text{err}} - p_{\text{err}}\|_\infty \le \epsilon.$$

As described in §1, the ideal sample complexity to strive for now is $O(\frac{1}{\epsilon^2 \eta})$.

Adaptivity, and a floor on η. Let us make two more technical remarks. First, if η is extraordinarily small (or even 0), we won't want to make $1/\eta$ measurements. Thus we assume the algorithm is given a floor η_0, and when $\eta \le \eta_0$ we are satisfied just to certify that this is the case. Second, we cannot hope to have (as before) a completely nonadaptive algorithm achieving sample complexity on the order of $1/(\epsilon^2\eta)$ because the algorithm does not know η, or even an approximation to η, in advance. Thus our algorithm will first need to find a

preliminary constant-factor approximation η_{est} to η in an online probe-and-measure fashion; then it can proceed nonadaptively.

5.1 Roughly estimating the error rate

Here we describe the (mildly) "adaptive" algorithm that handles the error floor and obtains η_{est}, a factor-5 approximation of η before subsequently finding a good approximation to all the error rates (including $p(0^n) = 1 - \eta$).

Lemma 24. *There is a randomized learning algorithm that, given input $0 < \delta_0, \eta_0 < 1$, as well as access to an n-qubit Pauli channel defined by distribution p with nontrivial error rate $\eta = 1 - p(0^n)$:*

- *repeatedly prepares a state, passes it through the Pauli channel, and measures, as in Theorem 1;*
- *halts after some number of repetitions (*always *at most $O(1/\eta_0) \cdot \log(1/\delta_0)$) and outputs either: "$\eta \leq \eta_0$" or else an estimate η_{est} that is within a factor of 5 of η;*
- *runs in classical time that is linear in the number of measurement readouts.*

Except with probability at most δ_0, the algorithm's output is correct and it halts after at most $O(1/\eta) \cdot \log(1/\delta_0)$ repetitions.

Proof. Recall Fact 20: by doing random nontrivial probes, an algorithm can get samples from a random string that is non-0^n with some probability η' between $\frac{2}{3}\eta$ and η. In order to find the factor-5 approximation η_{est} of η, it suffices for the algorithm to estimate η' up to a factor of 3 or else certify $\eta' \leq \eta_0$. This is now a completely standard problem: estimating the bias of an η'-biased coin up to a factor of 3 using on the order of $1/\eta'$ flips, despite not knowing η' in advance. The algorithm is the obvious one: repeatedly flip until getting "heads" (but never more than $O(1/\eta_0)$ times), convert the number of flips $\boldsymbol{G}$ into the estimate $1/\boldsymbol{G}$, then take the median of $O(\log(1/\delta))$ estimates. We omit the straightforward classical analysis. □

5.2 Individual Recovery with multiplicative error

We henceforth assume the algorithm from Lemma 24 succeeded and that η_{est} is a factor-5 approximation of the true error rate η. We now describe how the algorithm can do "Individual Recovery" with multiplicative error. A note: the sample complexities are stated in terms of the parameter η; formally, the algorithm does not know η, but it can use $5\eta_{\text{est}}$ (which it knows) in its place, and the $O(\cdot)$ bounds are not affected.

We first show that the algorithm from Theorem 22 already achieves the desired multiplicative-error/sample tradeoff in the case of estimating η:

Proposition 25. *Given η_{est} within a factor 5 of $\eta = 1 - p(0^n)$, a version of Theorem 22 holds in which, for $B = 0^n$, the estimate $\widehat{p}(0^n)$ satisfies $|\widehat{p}(0^n) - p(0^n)| \leq \epsilon\eta$ except with failure probability at most δ_0, and the number of samples used is $m = O(\frac{1}{\epsilon^2\eta}) \cdot \log(1/\delta_0)$.*

Remark 26. The success event here is equivalent to the estimate $\widehat{\eta} = 1 - \widehat{p}(0^n)$ satisfying the inequality $(1 - \epsilon)\eta \leq \widehat{\eta} \leq (1 + \epsilon)\eta$.

Proof. The algorithm used is the same as the one in Theorem 22 (with $B = 0^n$); only the analysis changes. Recall that the algorithm's estimate is the empirical mean of $\boldsymbol{H} = (-1/2)^{|\boldsymbol{R}|}$, a random variable whose true mean is $p(0^n) = 1 - \eta$. Equivalently we may consider the random variable $\overline{\boldsymbol{H}} = 1 - \boldsymbol{H}$, which has true mean η and which is supported in $[0, 2]$. But now a standard multiplicative Chernoff bound shows that the empirical mean $\widehat{\eta}$ of $\overline{\boldsymbol{H}}$ after $O(1/(\epsilon^2\eta)) \cdot \log(1/\delta_0)$ samples indeed satisfies $(1-\epsilon)\eta \leq \widehat{\eta} \leq (1+\epsilon)\eta$. □

Proposition 27. *A trivial modification of Theorem 22 also achieves, for any $B \neq 0^n$, an estimate $\widehat{p}(B)$ satisfying $|\widehat{p}(B) - p(B)| \leq \epsilon\eta$ except with failure probability at most δ_0, using $m = O(\frac{1}{\epsilon^2\eta}) \cdot \log(1/\delta_0)$ samples.*

Proof. Rather than empirically estimating the mean of $\boldsymbol{H} = (-1/2)^{|\boldsymbol{A}\star B + 2\boldsymbol{R}|}$, the algorithm instead empirically estimates the mean of $\boldsymbol{H'} = \boldsymbol{H} - (-1/2)^{|\boldsymbol{A}\star B|}$, a random variable bounded in $[-2, 2]$. (Note that Alice knows B and also each probe string $\boldsymbol{A}$, hence can compute $(-1/2)^{|\boldsymbol{A}\star B|}$ herself.) It is easy to see that $\mathbf{E}[(-1/2)^{|\boldsymbol{A}\star B|}] = 0$ using $B \neq 0^n$. Thus $\boldsymbol{H'}$ remains an unbiased estimator for $p(B)$; i.e., $\mathbf{E}[\boldsymbol{H'}] = p(B)$. But furthermore note that $\boldsymbol{H'}$ is almost always 0; specifically, whenever the channel outcome $\boldsymbol{C}$ is 0^n (probability $1 - \eta$), we have $\boldsymbol{R} = \boldsymbol{A} \star 0^n = 0^n$ and hence $\boldsymbol{H'} = (-1/2)^{|\boldsymbol{A}\star B|} - (-1/2)^{|\boldsymbol{A}\star B|} = 0$. Thus using $|\boldsymbol{H'}| \leq 2$ we trivially conclude $\mathbf{E}[(\boldsymbol{H'})^2] \leq 4\eta$. But now it follows from the Bernstein inequality (see, e.g., [28, Ch. 2, Prop. 2.4]) that to estimate the mean of a random variable $\boldsymbol{H'}$ that is bounded in $[-2, 2]$ and has $\mathbf{E}[(\boldsymbol{H'})^2] = s$, it suffices to use $\frac{s+2\gamma/3}{\gamma^2} \ln(2/\delta_0)$ samples to achieve additive error γ except with probability at most δ_0. Thus taking $\gamma = \epsilon\eta$ and using $s \leq 4\eta$ indeed completes the proof. □

5.3 Population Recovery with multiplicative error

Combining the results from the previous section on Individual Recovery with the reduction in §4.2 immediately proves our Theorem 2 (in the case of Pauli channels).

6 Further extensions: general channels and measurement noise

6.1 Pauli error rates of general channels

With very minor effort we can now upgrade our algorithm to learn the "Pauli error rates" of a *general* quantum channel, thereby fully establishing our Theorem 1.

We recall the following definitions/facts (see, e.g., [6, Lem. 5.2.4]):

Definition 28. Let Λ denote an arbitrary n-qubit quantum channel. Its *Pauli twirl* Λ_P is the n-qubit quantum channel defined by

$$\Lambda_P \rho = \mathop{\mathbf{E}}_{\boldsymbol{T} \sim \{0,1,2,3\}^n} [\sigma_{\boldsymbol{T}}^\dagger (\Lambda \sigma_{\boldsymbol{T}} \rho \sigma_{\boldsymbol{T}}^\dagger) \sigma_{\boldsymbol{T}}].$$

The channel Λ_P is itself a Pauli channel; the associated probabilities $p(C)$ are called the *Pauli error rates* of Λ.

Fact 29. Suppose we write K_j for the Kraus operators of Λ, so $\Lambda\rho = \sum_j K_j \rho K_j^\dagger$. Further suppose that K_j is represented in the Pauli basis as $K_j = \sum_{C \in \{0,1,2,3\}^n} \alpha_{j,C} \sigma_C$. Then Λ's Pauli error rates are given by $p(C) = \sum_j |\alpha_{j,C}|^2$.

It's easy to see that, given access to a general channel Λ, a learner Alice can simulate access to its Pauli twirl Λ_P: whenever Alice wishes to pass ρ through Λ_P, she instead chooses $\boldsymbol{T} \sim \{0,1,2,3\}^n$ uniformly at random, passes $\sigma_{\boldsymbol{T}} \rho \sigma_{\boldsymbol{T}}^\dagger$ through Λ, and replaces the channel output τ with $\sigma_{\boldsymbol{T}}^\dagger \tau \sigma_{\boldsymbol{T}}$.

In our context of learning Pauli error rates, this simulation becomes particularly simple. Recall that our algorithm for Pauli channels only ever passes pure states of the form $|\chi_+^{A_1}\rangle |\chi_+^{A_2}\rangle \cdots |\chi_+^{A_n}\rangle$ through the channel, for $A \in \{1,2,3\}^n$. Further, the channel output is always measured in the associated Pauli bases, the jth qubit of the output measured in the basis $|\chi_\pm^{A_n}\rangle$. The effect of simulating the Pauli twirl with $\sigma_{\boldsymbol{T}}$ is simply to replace the input $|\chi_+^{A_j}\rangle$ to qubit j with the input $|\chi_{(-1)^{A_j \star \boldsymbol{T}_j}}^{A_j}\rangle$, and to add $A \star \boldsymbol{T}$ to the measurement outcomes. Thus we may deduce the full version of Theorem 2 (concerning learning Pauli error rates of general channels) from the already-established special case of learning Pauli channels.

6.2 Tolerating measurement errors

It is also straightforward to see that our algorithm can tolerate a mild form of measurement error. Suppose that we have an imperfect 1-qubit measuring device that is used to implement the three Pauli-basis measurements. More precisely, we assume it has the following property: When applied to a qubit in a Pauli eigenvalue state, the measuring device "fails" (say, reads out "?") with probability ν, and otherwise behaves ideally. Here ν is a parameter that we assume is known to the learner through estimation, and that measurement failures are independent events.

As discussed in the paragraph just preceding §4.1, our algorithm for estimating any $p(B)$ is effectively performing the standard "Individual Recovery algorithm" for the *binary erasure channel* with erasure probability $\frac{1}{3}$. (Recall that we actually have the Z-channel with crossover probability $\frac{1}{3}$ applied to the binary string $C^{\neq B}$, but that the erasure channel algorithm only uses the locations of the 1's in the received string, and thus works equally well for the Z-channel.) The effect of measuring device failures is to replace the erasure probability $\frac{1}{3}$ with $r := \nu + (1-\nu)\frac{1}{3}$. So long as $r \leq \frac{1}{2}$, the standard recovery algorithm for probability r-erasures works just as well [9]: the only change needed is that the factor "$(-1/2)$" appearing in Theorem 22's definition of $\boldsymbol{H}$ needs to be replaced by $-r/(1-r)$. (Note that this quantity has magnitude bounded by 1 if and only if $r \leq \frac{1}{2}$.) But the condition $r \leq \frac{1}{2}$ is equivalent to $\nu \leq \frac{1}{4}$, and this justifies our Theorem 3.

(In fact, for erasure probability $\frac{1}{2} < r < 1$, much more sophisticated algorithms [7, 25] can succeed at Individual Recovery, at the expense of increasing the sample complexity from the order of $1/\epsilon^2$ to the order of $1/\epsilon^{2r/(1-r)}$; but for simplicity, we ignore pursuing this extension.)

7 An alternative, Fourier approach

Here we give an alternative algorithm for learning Pauli channels, using a perspective from Boolean Fourier analysis; see [24, Chaps. 1, 3] for background and notation.

For Pauli channels, the $\mathbb{F}_2$-Fourier transform relates the error rates and the channel eigenvalues. The Pauli operators themselves are the eigenvectors of a Pauli channel, and we can

easily compute the eigenvalue associated to σ_A using the relation $\sigma_A\sigma_C = (-1)^{A\star C}\sigma_C\sigma_A$ via

$$\sigma_A \mapsto \sum_{C\in\{0,1,2,3\}^n} p(C)\cdot\sigma_C\sigma_A\sigma_C^\dagger = \sum_{C\in\{0,1,2,3\}^n} p(C)\cdot(-1)^{\sum_{i=1}^n (A\star C)_i}\sigma_A = \lambda_A\sigma_A,$$

so that $\lambda_A = \mathbf{E}_{\boldsymbol{C}\sim\{0,1,2,3\}^n}\left[2^{2n}p(\boldsymbol{C})\cdot(-1)^{\sum_{i=1}^n (A\star\boldsymbol{C})_i}\right]$. This clearly resembles an $\mathbb{F}_2$-Fourier transform.

To make this connection more explicit, in the remainder of this section we will identify the elements of $\{0,1,2,3\}$ with their base-2 representations in $\mathbb{F}_2^2$. Let us use overline to denote the swapping operation on two bits; i.e., $\overline{a_1a_2} = a_2a_1$ for $a_1, a_2 \in \mathbb{F}_2$. We extend the notation n-fold to vectors $A \in \mathbb{F}_2^{2n} \cong (\mathbb{F}_2^2)^n$. (Equivalently, we have $\overline{0}=0$, $\overline{1}=2$, $\overline{2}=1$, $\overline{3}=3$, and we extend the notation coordinate-wise to $A\in\{0,1,2,3\}^n$.) Now define the *symplectic dot product*

$$\langle A, C\rangle = \overline{A}\cdot C = \sum_{i=1}^n (A\star C)_i \mod 2,$$

where $A\cdot C$ denotes the usual dot product on $\mathbb{F}_2^{2n}$. A Pauli channel eigenvalue is now equivalently written in two ways as

$$\lambda_A = \mathop{\mathbf{E}}_{\boldsymbol{C}\sim\mathbb{F}_2^{2n}}\left[2^{2n}p(\boldsymbol{C})\cdot(-1)^{\langle A,\boldsymbol{C}\rangle}\right] = \mathop{\mathbf{E}}_{\boldsymbol{C}\sim\mathbb{F}_2^{2n}}\left[2^{2n}p(\boldsymbol{C})\cdot(-1)^{\overline{A}\cdot\boldsymbol{C}}\right].$$

Let us write φ for the probability *density* (vis-a-vis the uniform distribution) associated to p; i.e., $\varphi(C) = 2^{2n}p(C)$. Then the Fourier transform $f = \tilde{\varphi}$ is given by

$$f(A) = \tilde{\varphi}(A) = \mathop{\mathbf{E}}_{\boldsymbol{C}\sim\mathbb{F}_2^{2n}}[\varphi(\boldsymbol{C})(-1)^{A\cdot\boldsymbol{C}}] = \mathop{\mathbf{E}}_{\boldsymbol{C}\sim p}[(-1)^{A\cdot\boldsymbol{C}}] = \lambda_{\overline{A}}. \tag{3}$$

Observe that f (and equivalently λ) are functions $f:\mathbb{F}_2^{2n}\to[-1,1]$ and that $p=\tilde{f}$. Such group character averages were considered in the context of quantum noise estimation in [17]. While we can talk interchangeably about the Fourier coefficients of the density φ and the channel eigenvalues λ (as they are related by $f(A)=\lambda_{\overline{A}}$), we will focus on f in what follows.

We see from Equation (3) that

$$f(A) = \mathop{\mathbf{E}}_{\boldsymbol{C}\sim p}[(-1)^{\langle\overline{A},C\rangle}] = \mathop{\mathbf{E}}_{\boldsymbol{C}\sim p}[(-1)^{\sum_t(\overline{A}\star C)_t}], \tag{4}$$

and as we now describe this means Alice can straightforwardly estimate $f(A)$ for any A of her choosing.

Let's extend Definition 12 of "nontrivial probe" to allow not just for $A\in\{1,2,3\}^n$ but any $A\in\{0,1,2,3\}^n$; we omit the adjective "nontrivial" in this more general case. To handle coordinates j where $A_j=0$, Alice can simply put any qubit $|\chi\rangle$ into the jth position of her state $|\psi_A\rangle$, ignore the jth position coming out of the channel, and automatically treat the jth readout bit as $\mathtt{0}$. In this way, Fact 13 still holds: for any probe $A\in\{0,1,2,3\}^n$ and any string $C\in\{0,1,2,3\}^n$ drawn by Charlie, the readout is $R = A\star C\in\{\mathtt{0},\mathtt{1}\}^n$. It follows that Alice can empirically estimate the right-hand side of Equation (4) by repeatedly probing the channel with $\overline{A}$ and averaging the following function of $\boldsymbol{R}$, the readout: $(-1)^{\sum_t \boldsymbol{R}_t}$. This yields $f(A)$ to additive precision ϵ with confidence at least $1-\delta$, using $O(1/\epsilon^2)\cdot\log(1/\delta)$ probes; we refer to this as "efficient estimation".

We now see that Alice has (noisy) query access to $f : \mathbb{F}_2^{2n} \to [-1, 1]$, and her goal is to estimate the large values of $p = \tilde{f}$. This task is highly reminiscent of the task solved by the Goldreich–Levin learning algorithm [13]. The minor differences are that Goldreich–Levin typically assumes *perfect* query access to some $f : \mathbb{F}_2^2 \to \{-1, 1\}$, and has the normalization that $\sum_C \tilde{f}(C)^2 = 1$, rather than our normalization of $\sum_C \tilde{f}(C) = \sum_C p(C) = 1$. Still, if one "unrolls" the Goldreich–Levin algorithm in this context, one gets almost the same solution for learning Pauli channels as described in §4.2: reduction from Population Recovery to Individual Recovery.

7.1 The Goldreich–Levin approach

In a typical exposition of the Goldreich–Levin algorithm (e.g. [24, Ch. 3.5], which we'll follow), one assumes Alice has perfect query access to an $f : \mathbb{F}_2^n \to \{-1, 1\}$. Herein we sketch the alterations to this exposition that are needed for learning Pauli channels. We note that a "quantum Goldreich–Levin" algorithm was given by Montanaro and Osborne [22] for learning the class of quantum boolean functions, which includes the unitary Pauli channels, but this makes explicit use of the unitary property and hence doesn't immediately apply to general Pauli channels.

One basic subroutine in the Goldreich–Levin algorithm (akin to "Individual Recovery") is using query access to f to efficiently estimate $\tilde{f}(B)$ for various B. This is done (see [24, Prop. 3.30]) via straightforward empirical estimation:

$$\tilde{f}(B) = \mathop{\mathbf{E}}_{\boldsymbol{A} \sim \mathbb{F}_2^{2n}} [f(\boldsymbol{A})(-1)^{\boldsymbol{A} \cdot B}]. \tag{5}$$

Recall that in our setting, Alice can only access $f(\boldsymbol{A})$ by empirically estimating it via Equation (4). Inserting this into the above, we get

$$\tilde{f}(B) = \mathop{\mathbf{E}}_{\boldsymbol{A} \sim \mathbb{F}_2^{2n}} \mathop{\mathbf{E}}_{\boldsymbol{C} \sim p} [(-1)^{\sum_t (\overline{\boldsymbol{A}} \star \boldsymbol{C})_t + \boldsymbol{A} \cdot B}].$$

Thus as needed in Goldreich–Levin, Alice can efficiently estimate this for any B of her choosing by picking uniformly random $\boldsymbol{A} \in \{0, 1, 2, 3\}^n$, probing the channel with $\overline{\boldsymbol{A}}$, and averaging the following function of $\boldsymbol{R}$, the readout: $(-1)^{\sum_t \boldsymbol{R}_t + \boldsymbol{A} \cdot B}$. Indeed the reader will note that this method is almost the same as the one used in Theorem 22! The essential difference is that $\boldsymbol{A}$ is uniform on $\{0, 1, 2, 3\}^n$ rather than $\{1, 2, 3\}^n$, which effectively makes the "crossover probability" $\frac{1}{2}$ instead of $\frac{1}{3}$, and hence the factor $(-1/2) = -\frac{1/3}{1-1/3}$ becomes $(-1) = -\frac{1/2}{1-1/2}$. Note that this difference implies that the Goldreich–Levin approach does not immediately tolerate measurement failures as in §6.2.

As mentioned earlier, Goldreich–Levin typically assumes $f : \mathbb{F}_2^n \to \{-1, 1\}$ and hence we have $\sum_{C \in \mathbb{F}_2^n} \tilde{f}(C)^2 = 1$; its goal is to find all B with $|\tilde{f}(B)| \geq \epsilon$, knowing that there are automatically at most $1/\epsilon^2$ such B. It accomplishes this via a "branch-and-prune" strategy that relies on the ability to estimate $\sum_{C' \in \mathbb{F}_2^{n-k}} \tilde{f}(\beta, C')^2$ for any prefix $\beta \in \mathbb{F}_2^k$. In our setup, with $f : \mathbb{F}_2^{2n} \to [-1, 1]$, we instead know a priori that $p = \tilde{f}$ satisfies $\sum_C \tilde{f}(C) = 1$, and our goal is to find all B with $|\tilde{f}(B)| \geq \epsilon$. Thus the search is even easier than in Goldreich–Levin, as the same branch-and-prune strategy works with non-squared Fourier coefficients. Following the strategy gives the same Population-to-Individual Recovery algorithm as in §4.2.

7.2 Final remarks

As mentioned earlier, the techniques used in the previous works [10, 14, 15] on Pauli channel estimation are Fourier-based. The paper [10] achieves SPAM tolerance, and manages to trade some measurement complexity for channel-reuse; on the other hand, its bounds have a dependency on the channel eigenvalue gap $\Delta = \min_{A \neq 0^n}\{1 - |\lambda_A|\}$, which may be arbitrarily small. As shown in the previous section, one can recover our (SPAM-less) Pauli estimation results via the Fourier approach with no dependence on Δ and without assumptions about the noise or the support.

We believe that it is possible to obtain a common generalization of the results in [10] and the present paper that achieves the best of both worlds via this Fourier approach: SPAM-robust and efficient Pauli channel estimation with no dependence on Δ. We leave this for future work.

Acknowledgements

We thank Robin Harper for discussions about Pauli channels. This work was supported by ARO grant W911NF2110001. R.O. is additionally supported by NSF grant FET-1909310. This material is based upon work supported by the National Science Foundation under grant numbers listed above. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation (NSF).

References

[1] F. Ban, X. Chen, A. Freilich, R. Servedio, and S. Sinha. Beyond trace reconstruction: Population recovery from the deletion channel. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science*, pages 745–768, 2019, arXiv:1904.05532.

[2] F. Ban, X. Chen, R. A. Servedio, and S. Sinha. Efficient Average-Case Population Recovery in the Presence of Insertions and Deletions. In D. Achlioptas and L. A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, volume 145 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:18, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, arXiv:1907.05964.

[3] L. Batman, R. Impagliazzo, C. Murray, and R. Paturi. Finding heavy hitters from lossy or noisy data. In *Proceedings of the 16th Annual International Conference on Approximation Algorithms for Combinatorial Optimization Problems*, pages 347–362, 2013.

[4] C. H. Bennett and S. J. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Phys. Rev. Lett.*, 69:2881–2884, Nov 1992.

[5] C. L. Canonne. A short note on learning discrete distributions, 2020, arXiv:2002.11457.

[6] C. Dankert. *Efficient Simulation of Random Quantum States and Operators*. PhD thesis, University of Waterloo, 2015, arXiv:quant-ph/0512217.

[7] A. De, R. O'Donnell, and R. Servedio. Sharp bounds for population recovery. *Theory of Computing*, 16(6):1–20, 2020, arXiv:1703.01474.

[8] A. De, M. Saks, and S. Tang. Noisy population recovery in polynomial time. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 675–684, 2016, arXiv:1602.07616.

[9] Z. Dvir, A. Rao, A. Wigderson, and A. Yehudayoff. Restriction access. In *Proceedings of the 3nd Annual Innovations in Theoretical Computer Science*, pages 19–33, 2012.

[10] S. Flammia and J. Wallman. Efficient estimation of Pauli channels. *ACM Transactions on Quantum Computing*, 1(1):1–32, 2020, arXiv:1907.12976.

[11] S. T. Flammia. PauliPopRec, Github repository, 2021.

[12] A. Fujiwara and H. Imai. Quantum parameter estimation of a generalized Pauli channel. *Journal of Physics A: Mathematical and General*, 36(29):8093–8103, jul 2003.

[13] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[14] R. Harper, S. T. Flammia, and J. J. Wallman. Efficient learning of quantum noise. *Nature Physics*, 16(12):1184–1188, Aug 2020, arXiv:1907.13022.

[15] R. Harper, W. Yu, and S. T. Flammia. Fast estimation of sparse quantum noise. *PRX Quantum*, 2(1):010322, Feb 2021, arXiv:2007.07901.

[16] M. Hayashi. *Quantum Information Theory*. Springer Berlin Heidelberg, 2nd edition, 2017.

[17] J. Helsen, X. Xue, L. M. K. Vandersypen, and S. Wehner. A new class of efficient randomized benchmarking protocols. *npj Quantum Information*, 5(1):71, Aug. 2019, arXiv:1806.02048.

[18] E. Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, mar 2005, arXiv:quant-ph/0410199.

[19] S. Lovett and J. Zhang. Improved noisy population recovery, and reverse Bonami–Beckner inequality for sparse functions. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 137–142, 2015.

[20] S. Lovett and J. Zhang. Noisy population recovery from unknown noise. In *Conference on Learning Theory*, pages 1417–1431, 2017.

[21] A. Moitra and M. Saks. A polynomial time algorithm for lossy population recovery. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 110–116, 2013, arXiv:1302.1515.

[22] A. Montanaro and T. J. Osborne. Quantum Boolean functions. *Chicago Journal of Theoretical Computer Science*, 2010(1):1–45, January 2010, arXiv:0810.2435.

[23] S. Narayanan. Improved algorithms for population recovery from the deletion channel. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1259–1278. Society for Industrial and Applied Mathematics, Jan. 2021, arXiv:2004.06828.

[24] R. O'Donnell. *Analysis of Boolean functions*. Cambridge University Press, 2014, arXiv:2105.10386.

[25] Y. Polyanskiy, A. T. Suresh, and Y. Wu. Sample complexity of population recovery. In S. Kale and O. Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1589–1618, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR, arXiv:1702.05574.

[26] B. Terhal. Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2):307, 2015, arXiv:1302.3428.

[27] J. ur Rehman and H. Shin. Entanglement-free parameter estimation of generalized Pauli channels. *Quantum*, 5:490, Jul 2021, arXiv:2102.00740.

[28] M. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

[29] J. Wallman and J. Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5):052325, 2016, arXiv:1512.01098.

[30] A. Wigderson and A. Yehudayoff. Population recovery and partial identification. *Machine Learning*, 102(1):29–56, 2016.