# Fast estimation of sparse quantum noise

Robin Harper,[1,*] Wenjun Yu,[2,*] and Steven T. Flammia[3]

[1] *Centre for Engineered Quantum Systems, School of Physics, University of Sydney, Sydney, NSW 2006 Australia*
[2] *Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China 100084*
[3] *AWS Center for Quantum Computing, Pasadena, CA 91125 USA*

(Dated: June 13, 2020)

As quantum computers approach the fault tolerance threshold, diagnosing and characterizing the noise on large scale quantum devices is increasingly important. One of the most important classes of noise channels is the class of Pauli channels, for reasons of both theoretical tractability and experimental relevance. Here we present a practical algorithm for estimating the $s$ nonzero Pauli error rates in an $s$-sparse, $n$-qubit Pauli noise channel, or more generally the $s$ largest Pauli error rates. The algorithm comes with rigorous recovery guarantees and uses only $O(sn)$ measurements, an $O(sn^2)$-time classical computation, and Clifford quantum circuits. Using data from an IBM 14-qubit device, we validate a heuristic version of the algorithm that trades off using $O(n^2)$ measurements in exchange for simplified Clifford circuits. These data show that accurate and precise estimation of the probability of arbitrary-weight Pauli errors is possible even when the signal is two orders of magnitude below the measurement noise floor.

## I. INTRODUCTION

Estimating noise in quantum computers will become increasingly important as we begin to move into the phase of testing quantum error correction on current noisy intermediate-scale devices [1]. Much current effort is focused on identifying methods that will continue to be applicable as the system size increases beyond the few qubit regime [2–8]. In such larger systems it is important to identify not only the errors that occur when qubits are operated singly or in small groups but also the additional errors that occur when the device is operated in a manner and in conditions that will be required for the implementation of error-corrected logical qubits. If we are able to characterize the noise and noise types (such as control errors, decoherence and crosstalk errors) in such a system then that will allow us to better diagnose and fix such errors, for instance by enabling calibration in the presence of crosstalk. Characterization of the noise will also allow the construction of tailored quantum error-correcting codes and decoders and customized fault-tolerant protocols designed to counteract the specific noise in the system. Such bespoke systems have been shown to be more effective than their generic counterparts [9–13].

While compressed sensing [14–19] extends the reach of process tomography [20] in the regime where quantum channels can be assumed to be sparse, it is still unable to achieve scaling past a handful of qubits. Even with the sparsity assumption, the Hilbert space of a multi-qubit machine makes it intractable to estimate all possible parameters in an efficient and fast way.

However, practical methodologies have been developed that can transform the noise in a device to noise that is well-approximated by a Pauli channel [21–23]. Such methods can be implemented without impacting the error rates of the device. This reduces the number of parameters of interest to $4^n$, where $n$ is the number of qubits of the device. In [24] it was shown how to sample these parameters in an efficient way and [25] showed how by averaging over a local Pauli basis the sampling could be made scalable. However it is clearly not scalable to estimate all $4^n$ parameters. Despite this apparent intractability, as we move towards quantum devices that are potential candidates for error-correction it is clear that our regime of interest becomes a machine where the relevant error rates, and in particular multi-qubit error rates, will be sparse and thus their estimation potentially scalable. Given this sparsity the question then becomes are we able to extract such sparse error rates in manner that is efficient and scalable with respect to the sparsity. In this paper we show this can be done.

Here we expand on the work presented in [24] to show how to efficiently extract all Pauli error rates, where one can make the assumption that the Paulis with errors above a threshold of interest ($\epsilon$) are $s$-sparse. In particular it is important to be able to recover any specific high weight (multi-Pauli) errors that might exist in the device. While it might be expected there will be $k$-nearest-neighbor correlations that will be influenced by the topology and underlying physics of the device itself, ideally any protocol used should also highlight and recover any unexpected long range correlations that might exist.

In Ref. [24], the authors essentially derived a variant of the Kushilevitz-Mansour algorithm [26] for learning decision trees via the Fourier spectrum and applied it to the case of Pauli channels. In the present work, we leverage more recent insights from the theory of sparse Fourier transforms [27] to derive nearly optimal algorithms for sparse Pauli channels. **SF:** return to this later...

* These authors contributed equally.

## II. PRELIMINARIES

The notation used in this paper is set out in detail in appendix A, however the following summary covers the relevant terms. A Pauli channel $\mathcal{E}$ acting on a quantum state $\rho$ is of the form $\mathcal{E}(\rho) = \sum_j p_j P_j \rho P_j$, where $p_j$ is the error rate associated with the Pauli operator $P_j$. The Pauli error rates $p_j$ form a probability distribution. These are closely related to, but distinct from, the Pauli eigenvalues, which are defined to be $\lambda_j = 2^{-n}\mathrm{Tr}\big(P_j\mathcal{E}(P_j)\big)$. Thus, when a state $\rho$ is subjected to the noisy channel $\mathcal{E}$, the Pauli error rate $p_j$ describes the probability of a multi-qubit Pauli error $P_j$ affecting the system. In contrast the Pauli eigenvalues describe how faithfully a given multi-spin Pauli operator is transmitted. The Pauli error rates $p_j$ and eigenvalues $\lambda_j$ are related by a Walsh-Hadamard transform (where the transform is ordered by Pauli commutation relations — see appendix A for a further discussion). For any natural number $N$, then we write $[N]$ to mean $\{0, \ldots, N-1\}$.

In an analogy with discrete Fourier transforms, the Pauli error rates can be thought of as the frequency or spectral domain components of the time domain signal, being the Pauli eigenvalues. Here we wish to sparsely sample the dense time domain signal (the eigenvalues) and reconstruct the entire (but sparse) frequency domain (the error rates).

The underlying assumption is that for a quantum channel on $n$ qubits we have a sparse underlying Pauli error rate. Here we are going to assume there are only $s \ll 4^{n/2}$ Pauli error rates with probabilities greater than $\epsilon$ in this distribution. This is what we mean when we refer to an $s$-sparse model.

Under the $s$-sparse assumption, we show how to estimate all the error rates greater than $\epsilon$ with high accuracy with polynomial sampling and time complexity.

Our recovery methodology builds on the main result of Ref [24] and an adaptation of the classical algorithms described in Refs [27, 28]. In Ref [24] Flammia and Wallman show how to recover the Pauli fidelities of a stabilizer covering (i.e. $2^n$ Paulis) to *relative* precision $\epsilon$ using $O\big(\epsilon^{-2}n\big)$ measurements. The circuit modifications required to be made to that algorithm are shown in Figure 1. The recovery of all $4^n$ Pauli fidelities would require $2^n$ applications of this algorithm, which is clearly not scalable in $n$. While Ref [24] presents a binary search algorithm for high-weight Pauli errors that is efficient in the number of Pauli errors that need to be determined, numerical reconstructions confirm that the number of Pauli eigenvalues that have to be determined are close to the theoretical bounds of $O(\frac{1}{\epsilon^2}\log(\ldots))$ Pauli eigenvalues per Pauli error rate and consequently it would prove difficult to use in practice.

We show how the $s$-sparse Pauli error rates can be ascertained by making a practical and scalable number of queries of Pauli eigenvalues. We will assume the Pauli eigenvalues are determined using the algorithm from [24] (although they can, in principle, be determined by using

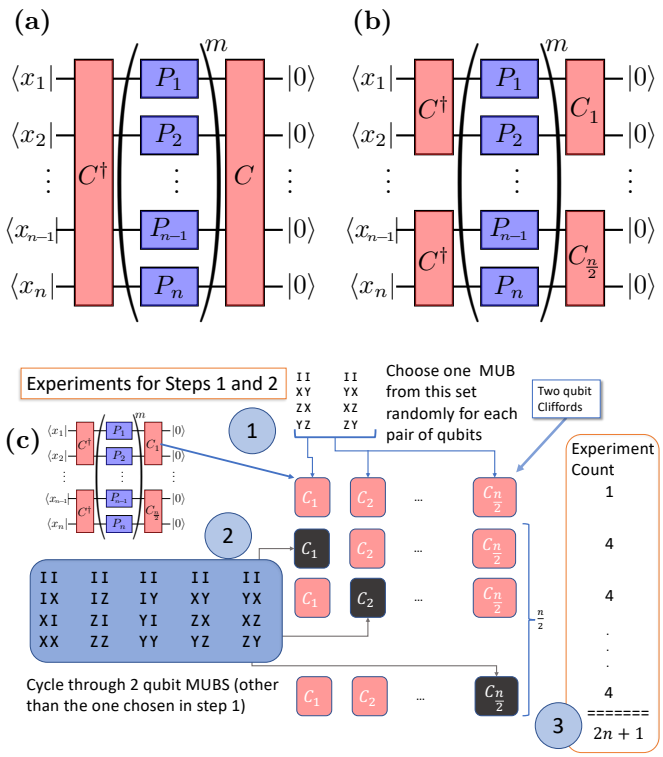



FIG. 1. **(a)** shows the protocol (slightly modified) as described in [24]. A single $n$-qubit Clifford is used to select a random $n$-qubit stabiliser set. Paulis are then used to twirl the channel, before it is returned to the computational basis for measurement. The procedure is repeated for various values of $m$. The Cliffords determine which of the $2^n$ Pauli fidelities are recovered by the circuit. **(b)** shows a further modification of the circuit, where instead of using a generic $n$-qubit Clifford only two-qubit Cliffords are used. (Where the device has an odd-number of qubits a single Clifford can be used on one of the qubits.) Since there are only random Paulis in the sequence between the Cliffords, a further two qubit Clifford per pair is sufficient to return the computation to the computational basis. As discussed in the text for each chosen value of $m$, the circuit is repeated for multiple sequences with different randomly chosen Paulis, but for fixed Cliffords. Collectively each of the runs for multiple-choices of Paulis carried out over several different lengths of m are knows as *an experiment*. **(c)** shows how once an experiment has been chosen for Step 1 of the procedure (1), further experiments are created in order to provide the information needed to provide the offsets required to identify the Pauli (see text). As shown in (2) each of the two qubit Cliffords needs to be cycled sequentially through the four other 2-qubit stabiliser groups (i.e. the four that are different from the initial choice). This means that for each sequence in Step 1, a further $2n$ experiments needs to performed (3), leading to $2n+1$ experiments per stabiliser group chosen. For the second group an offset of one qubit should be chosen, meaning the local stabiliser groups now span different qubit pairs. Simple Pauli twirls can be carried out on any odd or isolated qubits.

any other appropriate method). We show how to form queries that are efficient not only in $s$ but also in the number of experiments required to recover all $s$-sparse Paulis, delivering a mechanism that is efficient and easy to implement experimentally.

## III. ESTIMATING SPARSE PAULI RATES FROM NOISY FIDELITY

We will adapt and expand the work of Refs [28] and [27] (the reconstruction algorithms) to the current circumstance. The problem looked at by Scheibler et al. [28] and later (in the regime of noisy signals) by Li et al. [27] involved decoding a signal $x \in \mathbb{R}^N$, which contains $N = 2^n$ samples, indexed by $m$ a binary index i.e, where $m \in \mathbb{F}_2^n$.

In our circumstances we are not analysing the frequency domain of a signal, but rather the global probability distribution of the Pauli errors in a quantum device and the eigenvalue distribution of the Paulis in a super-operator representation of the averaged noise channel. For an $n$-qubit device we have $N = 4^n$, being the number of distinct Pauli eigenvalues, which requires the index $m$ to be $2n$ bits long i.e. $m \in \mathbb{F}_2^{2n}$. We discuss the index we use shortly.

As discussed above, for a particular n-qubit Pauli $P_j$, the error rate associated with that Pauli is $p_j$ and the eigenvalue of that Pauli (representing how faithfully that multi-spin Pauli can be transmitted through the averaged noise channel in the device) is $\lambda_j$.

In this case we have a Walsh Hadamard transform coefficient, computed as:

$$\lambda_k = \sum_{m \in \mathbb{F}_2^{2n}} (-1)^{\langle k,m \rangle} p_m, \tag{1}$$

where $\langle k, m \rangle$ represents the inner product of the Paulis represented by these bistrings. The symmetrical nature of the Walsh-Hadamard transform (including this variation, see appendix A for more details) means we also have:

$$p_m = \frac{1}{N} \sum_{k \in \mathbb{F}_2^{2n}} (-1)^{\langle m,k \rangle} \lambda_k. \tag{2}$$

To relate this to our problem we note that each $n$-qubit Pauli can then be represented by a binary string $2n$ bits long. We imagine expressing each Pauli as a bit string, mapping $I \to 00, X \to 01, Y \to 10,$ and $Z \to 11$.

Given this mapping the algorithms presented in [28] and [27] are broadly applicable (we will note where adjustments have to be made) and of course we wish to exploit our ability to make $2^n$ simultaneous commuting measurements.

Below we give a broad overview of the reconstruction algorithms as applicable to our needs. For specific details the reader is referred to Ref [27] and for reconstruction guarantees in our specific case, to Theorem 1. We first deal with the noiseless case.

The main idea behind the algorithm is to note that each Pauli eigenvalue is made up of a linear combination of all the Pauli error rates. By sub-sampling the eigenvalues, we are able to split up the Pauli error-rates, figuratively creating buckets of error rates, where each 'bucket' contains a linear combination of a smaller number of error rates. Provided that there are sufficient buckets, then in the sparse regime most of these buckets will only contain a few Pauli error rates with weight $\geq \epsilon$. If we can identify these buckets we can evaluate these error rates and this information will allow us to reconstruct all the sparse error rates. With this in mind, the reconstruction algorithm can be broken down into three main steps:

1. Determining the sub-sampling bins and performing experiments to sample the required eigenvalues.

2. Construction of the correct offsets (and subsequent sampling) to enable identification of single Pauli-bins (*single-tons*) and the Pauli which occupies them.

3. Running the decoder, to 'peel back' single-tons, converting multi-Pauli bins to single-Pauli bins.

We describe these three steps in an intuitive manner below. Proofs as to the efficacy of the algorithm described are contained in Appendix D.

**Step 1**. The intuition behind the first step is that it is possible to sample a specific pattern of eigenvalues that will allow the reconstruction of the global probability vector, but where various probabilities are binned (i.e. added together). For instance, given a global probability vector with $4^n$ values it is possible to rewrite this as a quasi-probability vector ($\tilde{p}$) with $2^n$ values, each value being composed of the sum of $2^n$ of the original global probability values. In the regime where our sparsity is $< 2^n$ then we will show that with appropriate random sampling a large number of these quasi-probability vector values (which we will call buckets), will be composed of none or one of our sparse Pauli errors, i.e. those with a weight $\geq \epsilon$.

Whereas [28] imagined using specific bit patterns of binary strings to index the requisite eigenvalues to sample, we wish to exploit the ability of a quantum device with independent measurement on each qubit to sample from a bit string of $2^n$ values. As previously discussed the protocol in [24] shows how to measure, to multiplicative precision, the Pauli eigenvalues of $2^n$ commuting Paulis using one randomized-benchmarking style experiment. The constraint that the Paulis measured be mutually commuting is exactly the constraint we require for the sub-sampling to allow us to create the required quasi-probability vector.

So let us assume we have a randomly chosen stabiliser group, stabilising an $n$-qubit state. The stabiliser is a linear subspace of $\mathbb{F}_2^{2n}$ (using our representations of Paulis)

such that $\langle a, b\rangle = 0$ for all $a, b \in \mathcal{S}$, with $2^n$ group elements. The group itself will consist of the trivial Pauli ($P_I^{\otimes n}$) and $2^n - 1$ non-trivial Paulis.

So let $\Psi_j$ represent the $j^{\text{th}}$ element of the chosen stabiliser group. Our quasi probability vector (consisting of $2^n$ bins each containing $2^n$ distinct Pauli errors) becomes:

$$\tilde{p}_j = \frac{1}{2^n} \sum_{l \in \mathbb{F}_2^n} \lambda_{\Psi_l} (-1)^{\langle j,l \rangle}, j \in \mathbb{F}_2^n \,. \tag{3}$$

The effect of this is that $2^n$ sampled Pauli eigenvalues (that is the eigenvalues relating to the Paulis forming the stabiliser group), when transformed by the Walsh Hadamard transform, gives us $2^n$ bins, each containing a sum of $2^n$ error rates.

The first assumption of the algorithm then becomes that there will be a number of bins that only contain one Pauli error rate with a weight $\geq \epsilon$ (the other Pauli errors allocated to that bin being, effectively, zero).

This will depend on the size of the bins and the sparsity of the Pauli error rates, and is discussed further in Appendix D.

So how do we construct our stabiliser group? The most obvious way is to sample a random $n$-qubit Clifford (see [29] for how to do this). However as $n$ grows past a few qubits, then on current devices the number of single and multi-qubit gates required to construct such a Clifford circuit will run the risk of flattening the signal required to estimate the eigenvalues. A better way for current devices is to use a random subset of $n$-qubit stabilisers that can be formed from a single round of non-overlapping 2-qubit Clifford gates. This has the added advantage of making it trivial to work out how to perform step 2.

**Step 2**. The question then becomes how do we detect which bins contain a single Pauli error rate? To do this the reconstruction algorithms use the shift/modulation property of the Walsh-Hadamard transform, specifically:

$$\lambda_{m+p} \overset{\text{WHT}}{\longleftrightarrow} (-1)^{\langle p,k \rangle} p_k \,. \tag{4}$$

By taking each element of the stabiliser group and offsetting the sample with a shifting bit pattern (e.g. for four qubits, the sample would be offset by the five following bit patterns $[0,0,0,0]$, $[1,0,0,0]$, $[0,1,0,0]$, $[0,0,1,0]$, $[0,0,0,1]$) then the Pauli error rates consigned to that bin are no longer merely summed but rather are added or subtracted depending on whether the inner product of their 'bit-strings' and the relevant pattern is zero or one. This result will be illustrated in more detail in Algorithm 2 and Lemma 1, where we also discuss how to use bit-flip error dectection codes to make the decoding more robust to noise.

$$\tilde{p}_{j,d} = \frac{1}{2^n} \sum_{l \in [2^b]} (-1)^{\langle j,l \rangle} \lambda_{(d+\Psi_l)}, j \in \mathbb{F}_2^n, d \in \{2^0 \ldots 2^{2n}\} \,. \tag{5}$$

This leads to a number of remarkable effects. If the bin is empty (i.e. contains no Pauli error rates with non-zero errors) each of the *offset bins* (i.e. for a particular $j$ each $\tilde{p}_{j,d}, d \in \{2^0 \ldots 2^{2n}\}$) will also be zero. If the bin contains only one non-zero Pauli error rate then the magnitude of the sum of each of the offset bins will be constant, and the sign of the sums will identify exactly which Pauli has the non-zero error rate. (For example using the four qubit offsets shown above, if the absolute values of the bins were all 0.001 and the signs of the 4 offset bins were $+--+$, this could only be caused by a single Pauli error rate of 0.001, with a bit string of 0110). In every other case, the bin contains multiple Pauli error rates (a *multi-Pauli bin*), which leads us to the PEELING decoder (see step 3).

So how can we construct the experiments that will allow us to extract the 'shifted' eigenvalues? For instance, one might note that for any particular stabilizer group S, the offset bit pattern applied to each of the elements of the group are unlikely to form a stabiliser group.

It transpires that where we use a stabiliser group created by local two-qubit Cliffords (on each qubit pair), we can do this simply by iterating each distinct qubit pair through four further (different) two-qubit stabiliser patterns. There are five two-qubit stabiliser groups in total, labelled $S_{1\ldots5}^{\otimes 2}$. We set these groups out in detail, together with the two-qubit circuit needed to create them in fig. 4. The initial stabiliser is chosen by selecting randomly from $S_{1,2}^{\otimes 2}$ for each qubit pair. This becomes the stabiliser for the purpose of Step 1. This circuit is used to conduct the first experiment and extract $2^n$ Pauli eigenvalues. The offset pattern required for this step 2, is constructed by iterating over each qubit pair, and replacing the circuit chosen in Step 1, with one of the other 5 (for a total of 4 further experiments per qubit pair). The total number of experiments required is therefore $2n + 1$. By analysing each of the experiments formed we will be able to pull out of the all the eigenvalues determined by such experiments. Figure 1 shows the circuits used for each experiment and illustrates the method described above.

**Step 3**. If we use a variety of sub-sampling matrices (that is we repeat Steps 1 and 2 for more than one random initial choice of Cliffords) we are now in the position where we have identified a number of Pauli error rates (from bins that contain only one Pauli error rate) and we will also have a number of bins that contain more than one Pauli error rate (multi-Pauli bins). For any two different stabiliser groups, different Pauli error rates will have been hashed into different buckets. Where we have identified a single Pauli error rates under, say, stabiliser group 1, that same error rate may be in a different bucket under stabiliser group 2, a bucket it shares with one or more different high weight Paulis (i.e. it may be in a multi-ton bucket under stabiliser group 2). However, because we know the value of this Pauli error rate (since it was a singleton under stabiliser group 1), we can remove it from the bucket created by stabiliser group 2. The hope being that after this removal we are left with a bucket that now has only one Pauli error rate left in it. This removal of the value of a previously identified sin-

gleton from a different stabiliser group's bucket is known as 'peeling back' the known values – giving the PEELING decoder its name. The goal being that when we peel back our identified error rates, we might create more bins that now contain only one Pauli error rate. This can be applied in an iterative fashion until we have identified all Pauli error rates (all the bins are empty) or have no further single Pauli error rates to peel back, and all these steps can be viewed in Algorithm 3. In the latter case the reconstruction algorithm has failed, although we will know the magnitude of the error rates we have failed to identify.

**Dealing with noise**. Ref [27] shows how to adapt the reconstruction algorithm to deal with noise, where it is assumed the noise is of the form

$$p_m = \frac{1}{N} \sum_{k \in \mathbb{F}_2^{2n}} (-1)^{\langle m,k \rangle} \lambda_k + \epsilon_m \,, \tag{6}$$

where $\epsilon_m \sim \mathcal{N}(0, \sigma_m^2)$.

The 'signal' noise in our case is the error in our eigenvalue reconstructions caused by finite sampling. These finite sampling errors occur because of the limited number of sequences and measurement 'shots' per sequence occurring when the experiments are carried out. These errors are analysed and bounded for the present protocol in [24], where they show how the reconstruction error rates are multiplicative, that is the percentage error rate remains constant even as the the sum of 1 - the Pauli eigenvalue approaches 0. To reduce noise, the number of sequences and shots per experiment needs to be increased. However, despite the fact that the noise comes from such sources, we can assume that the errors from the least square fitting procedure used to determine the eigenvalues are normally distributed and thus that the noise is normally distributed. Should it be required the variance of the eigenvalue reconstruction can be sufficiently determined by bootstrapping the protocol from the observed measurement outcomes (this is described in detail in [25]).

The PEELING decoder only requires two adjustments to account for noise, namely the zero Pauli verification and the single Pauli search protocols.

For the former, under the exact model we identify a bin as being empty if the value of the bin (and each of the offset bins) is zero. Where we have noise, we simply relax the requirement that the bins are exactly equal to zero before identifying them as empty. We can bound an acceptable small value as indicating an empty bin, given the number of 'noisy' zeros in the bin and our estimate of the noise. This will lead to a floor of Pauli error weights we can recover, i.e. we are unlikely to recover those Pauli errors with a value so small they are swamped by the noise in the bins. This is an inevitable consequence of the noise.

The single Pauli identification process has two aspects that need to be considered. The first is 'does the bin contain only a single Pauli?', the second is 'if so: which Pauli?'. For a noisy oracle the first question is dealt with the same way as the noisy zero, i.e. we only require the magnitudes of the offset bins to match to within some noise parameter. While this runs the risk of not noticing some small Pauli error rates that are also in the bin, it appears to work well in practice. The second is more akin to a noisy bit flip channel, in that the noise may cause us to incorrectly identify a '1' as a zero or vice-versa. (This is more likely when the noise is commensurate with or greater than the Pauli error weight.) One simple method of dealing with this is to repeat sample with different offsets, and then take a majority vote (see [27] for more details), however our numerical simulations do not suggest that this is necessary. Finally we can use a number of random offsets and some additional fixed offsets chosen in such a way they form a classical error correction code to further protect the algorithm from noise. While this does not alter the sample scaling, it will impact the number of experiments but it comes with a robust recovery guarantee and forms the basis for the recovery guarantee described in the next section and appendix D.

## IV. RECOVERY GUARANTEE FROM NOISY PAULI EIGENVALUES

Using the algorithm illustrated above and leveraging the proofs contained in [27], we can construct the recovery guarantee shown below relating to our ability to recover Pauli error rates with bounded error from noisy Pauli eigenvalues. Moreover, the necessary sampling and computing complexities are both polynomial with the sparsity $s$ and number of qubits $n$. The intuition behind the guarantees is that by increasing the number of offset observations we can induce an exponential decay in the the chance of the bin detector subroutine failing. If the bin detection succeeds, then as shown in [27] an oracle-based peeling decoder will succeed with high probability. The error bound arises through a suitable choice of the gap between a noisy result and increasing resolution of bin detector as the number of offset observations increase.

**Assumption 1.** Let $\mathbf{p} \in \mathbb{R}^{4^n}$ be the WHT coefficient vector with support $\mathcal{K} = \text{supp}(\mathbf{p})$. We make the following assumptions:

- **A1** The support set $\mathcal{K}$ is chosen uniformly at random from all subsets of $[N]$ of size exactly $s$. The sparsity $s = |\mathcal{K}| = O(4^{\delta n})$ is sub-linear in the dimension $4^n$ for some $0 < \delta < 1$. (see appendix C for a discussion of this assumption)

- **A2** Sufficient sequences are sampled that noise arising from an imperfect estimation of Pauli eigenvalues can be modelled by an identical normal distribution centered around the eigenvalue with variance $\xi^2$.

- **A3** Each coefficient $p_m$ for $m \in \mathcal{K}$ is lower bounded by $\rho_0$, and $\rho_0 \geq \sqrt{4\nu^2}$ where $\nu$ is the standard devia-

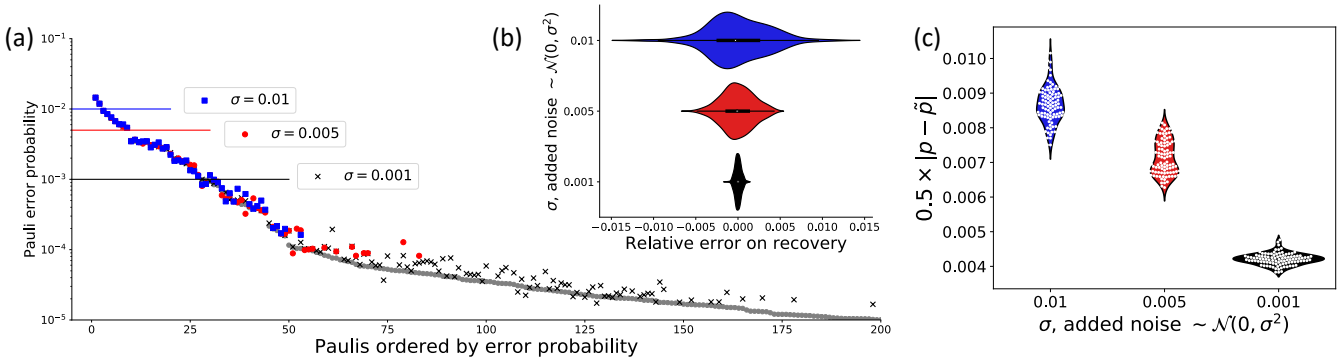



FIG. 2. (a) This figure shows the ability of the reconstruction algorithms to recover experimentally inspired sparse Pauli distributions. The distributions were from a 14 qubit device, and therefore the oracle consisted of $2^{28} = 268,435,456$ potential eigenvalues. The text (section V) contains a full description as to how the eigenvalue oracles were constructed. Varying levels of normally distributed noise $\sim \mathcal{N}(0, \sigma^2)$ was added to the oracle, and the algorithm was run to try and recover all the different Pauli eigenvalues. The scatter plot shows the accuracy with which the Paulis in the original distribution (with a weight $\geq 10^{-5}$) were recovered for various levels of noise. In all cases the eigenvalues sampled consisted of those retrieved via 58 random benchmarking style experiments. As a separate experiment 4 different randomly generated many body Paulis with error rates normally distributed around $\mu = 0.005, \sigma = 0.001$ were inserted into the probability distribution and the algorithm was run in an attempt to recover these Paulis. The relative errors in recovery were ascertained over 100 runs of this experiment and are plotted as (b), inset in the main graph. As can be seen these Pauli error rates were recovered with high relative precision in all cases. Finally the one norm trace distance between the original probability distribution ($p$) and the reconstructed probability distribution ($\tilde{p}$), being $0.5 \times |p - \tilde{p}|$ was calculated for 100 different randomly generated samples of noise and plotted in the right hand chart (c). As can be seen the entire probability distributions are consistently recovered to high precision.

tion of the unit bin error induced by the imperfect eigenvalues.

**Theorem 1.** *Let Assumption 1 hold for the target Pauli eigenvalues $\widehat{\lambda}$ and the corresponding Pauli error rates* $\mathbf{p}$. *Then for any sparsity regime $s = O(4^{n\delta})$ with $0 < \delta < 1$, the above algorithm computes the $s$-sparse Pauli error rates $\widehat{\mathbf{p}}$ with a noise bound $||\widehat{\mathbf{p}} - \mathbf{p}||_\infty \leq \sqrt{4\nu^2}$, where by retaining a sample complexity that scales only as $O(sn)$ and with a computation complexity that scales as $O(sn^2)$ the algorithm achieves a vanishing failure probability $\mathbb{P}_F$ for sufficiently large $n$.*

*Proof.* See appendix E □

## V. EXPERIMENTALLY INSPIRED NUMERICAL VERIFICATION OF THE PROTOCOL

In order to demonstrate the efficacy of the algorithm we have used data extracted from an IBM device. In Ref. [25] the full locally averaged Pauli error rates were extracted from a 14 qubit IBM device. Using this data set, which consists of $2^{14}$ locally averaged eigenvalues, we randomly projected the averaged noise onto the full Pauli probability simplex in order to simulate a full Pauli probability distribution. This reconstructed distribution involves an arbitrary, random, splitting of averaged data into its constituent components, although if re-averaged would match the experimentally derived data. In this sense, it is a realistic, experimentally inspired distribution. The probabilities were then transformed (using the Walsh Hadamard transform) to give the noise-free eigenvalue oracle.

The data used had an error free probability of approximately 86%, with approximately 200 Paulis having an error rate above $10^{-5}$, 600 above $10^{-6}$ and 2,000 above $10^{-8}$. In this part of the paper we were principally interested in the first regime i.e. above $10^{-5}$. In [27, 28] the sparsity of the error rates is identified by a number $\delta$, where the number of error rates $s$ is related to the potential number of error rates ($4^n$ for an $n$-qubit device) as $s \approx (4^n)^\delta$. The regime numerically explored in this part of the paper relates to a $\delta \approx 0.25$.

As can be seen from Figure 2 the sparse recovery protocol performs well in this regime ($\delta \leq 0.25$), requiring only a fraction of the eigenvalues that would be required for a full recovery of all Pauli error rates. The limiting factor in this regime is the noise in the oracle, which equates directly to the number of measurements/sequences sampled as part of the original measurement error (see [24] for relevant reconstruction guarantees). It appears that the effect of the protocol is to allow recovery of the Pauli error rates to, approximately, an order of magnitude less than the noise in the oracle. Importantly where there are many body Pauli errors 'hiding' in the distribution the protocol reveals and evaluates such Pauli error rates to a high degree of relative accuracy (Figure 2(b)).

Appendix B discusses the regime where ($\delta > 0.25$). In that case continued recovery of Paulis with low error

rates requires some changes to the local stabiliser groups used (or a switch to global random stabiliser groups).

## VI. DEVICES $\gg 30$ QUBITS

Here we address the practicality of the protocol once system size becomes greater than, say, 30 qubits. The protocol executed on the device remains identical as system size increases, however, each experiment returns sufficient information to reconstruct $2^n$ eigenvalues, which when $n$ is $> 30$ might prove problematic to store in a typical computer's working memory and to manipulate. All, however, is not lost.

One can take advantage of the fact that the Hadamard-Walsh transform commutes the marginalisation of the observed probabilities and the process of fitting required to ascertain the SPAM free eigenvalues (see [24]). The actual observations only require $n$ bits of data to store. We can, therefore, marginalise the observations to obtain overlapping sets of $2^m$ eigenvalues (where we choose $m$ to be the largest computationally tractable number for our classical computer). This will mean that we have multiple sets of $2^m$ buckets, each potentially containing $2^{2n-m}$ Pauli error rates. Given this, the $s$-sparse assumption now becomes $s < 2^m$. However, in systems of this size, where this is not the case, then most of the error rates will be extremely small.

## VII. CONCLUSION

We have shown that for sparse Pauli channels we can learn all Pauli errors, even those with associate with high weight Paulis, using realistic experimental resources that scale with the sparsity of the Pauli errors. In particular we have demonstrated that using only a few local two qubit gates and a number of quantum experiments that scale linearly (with a factor of about 4), we can recover upto $4^{\delta n}$ of the highest error Paulis, where $\delta \leq 0.25$. Our numerical analysis indicates in the regime where $0.25 \leq \delta \leq 0.5$ we can still recover these errors with a number of experiments that only scale quadratically in the number of qubits.

We back these experimental protocols with rigorous performance guarantees that confirm, with physically plausible assumptions, we only require to sample a number of Pauli eigenvalues that scale with $O(sn)$, where s is the assumed sparsity of the Pauli error probabilities in question. This coupled with our ability to utilise the protocols presented in [24] and [25] to learn up to $2^n$ commuting Pauli eigenvalues per experiment gives us the ability to use the protocol in a way that scales with experimental resources as described above.

This work provides an experimentally realisable method of identifying the relevant Pauli errors in large scale quantum devices, even if there are unexpected long range correlations between the qubits. The ability to do so will be vital as we seek to mitigate the errors in such devices, to learn the noise patterns that exist when such devices are operated holistically and will allow better designing and tailoring of error correction protocols in such devices.

### Appendix A: Notation and Preliminaries

Given a set of $n$ qubits with Hilbert space dimensions $d = 2^n$, we can introduce the following notation. Let $\mathbb{P}^n$ denote the group of Pauli operators on all $n$ qubits and $\mathbf{P}^n = \mathbb{P}^n/\langle i \rangle$. There is a natural isomorphism between multiplication on $\mathbf{P}^n$ and bit-wise addition of $2n$-bit strings $\mathbb{Z}_2^{2n}$ given by

$$a \in \mathbb{Z}_2^{2n},\ a \longleftrightarrow P_a = P_{a_x a_z} = i^{a_x \cdot a_z} X[a_x] Z[a_z], \quad \text{(A1)}$$

where $a_x, a_z \in \mathbb{Z}_2^n$ and $X$ and $Z$ are the standard single-qubit Pauli matrices, and $P_a \in \mathbb{P}^n$ is understood to be a canonical coset representative. Here $X[a_x] = X^{a_{x_1}} \otimes \ldots \otimes X^{a_{x_n}}$, and similar for $Z[a_z]$. Using this isomorphism, we can directly use $a \in \mathbb{Z}_2^{2n}$ to denote the Pauli matrix $P_a$. For any two Pauli matrices $P_a$ and $P_b$, we have $P_a P_b = (-1)^{\langle a,b \rangle} P_b P_a$ where

$$\langle a, b \rangle = a_x \cdot b_z + a_z \cdot b_x \mod 2 \quad \text{(A2)}$$

is symmetric and bilinear.

In this paper we use a variant of the Walsh Hadamard transform, where the ordering is determined by the commutation relations between the Paulis (represented as bit strings). The natural ordering of a Walsh Hadamard transform matrix can be calculated from the Kronecker product of a binary matrix, thus:

$$\text{WHT-natural ordering } = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes n} \quad \text{(A3)}$$

In this case, like the sequency order and dyadic order variants of the Walsh Hadamard transform we reorder the columns of the transform matrix. Unless otherwise expressly noted, we use a WHT where the $(i,j)$th entry of the Hadamard transform matrix is given by $(-1)^{\langle i,j \rangle}$ where the inner product is calculated using the commutation relation of the Paulis represented by the bit-strings $i$ and $j$. The advantages of using this variant of the WHT is that when it is used to transform Pauli eigenvalues to Paulis probabilities and vice-versa (eq. (1) and eq. (2)), the position of the Paulis in the various transformed vectors remain constant.

We define a *stabilizer group* $\mathbf{S}$ to be a linear subspace of $\mathbb{Z}_2^{2n}$ such that for all $a, b \in \mathbf{S}$, $\langle a, b \rangle = 0$. Thus a stabilizer group forms a commuting subgroup of the full Pauli group. Moreover, for a set $X \subseteq \mathbf{P}^n$, we can define a *stabilizer covering* $O$ which consists of stabilizer groups such that $O = \{S_j\}$ and $X \subseteq \bigcup_j S_j$.

We define the *commutant* of a set $G \subseteq \mathbf{P}^n$ to be a group $C_G \leqslant \mathbf{P}^n$ which satisfies

$$C_G = \{a \in \mathbf{P}^n : \forall g \in G,\ \langle a, g\rangle = 0\}. \tag{A4}$$

And we can further define the $anti-commutant$ $A_G$ of $G \subseteq \mathbf{P}^n$ to be the quotient group $\mathbf{P}^n/C_G$.

### 1. Quantum Channels and Model Assumptions

For a general linear quantum channel $\mathcal{L} : \mathbb{C}^{d\times d} \to \mathbb{C}^{d\times d}$, it can be represented in Kraus operator form:

$$\mathcal{L}(\rho) = \sum_k A_k \rho B_k^\dagger. \tag{A5}$$

In this paper we are interested in a specifiy type of channel, namely a *Pauli channel*. A Pauli channel $\mathcal{E}$ acting on a state $\rho$ can be represented as

$$\mathcal{E}(\rho) = \sum_a p_a P_a \rho P_a, \tag{A6}$$

where $p_a$ is the error rate associated with Pauli operator $P_a$. Moreover, we are more interested in a specific class of operations that is trace preserving channel, and a trace preserving Pauli channel must satisfies $\sum_{a\in\mathbf{P}^n} p_a = 1$ and consequently the set $\{p_a\}$ forms a probability distribution. These Pauli rates are closely related to the *Pauli eigenvalues* $\{\lambda_j\}$ where $\lambda_j = 2^{-n}\mathrm{Tr}[P_j\mathcal{E}(P_j)]$. In fact, the Pauli eigenvalues describe how faithfully the Pauli channel transmits a particular Pauli.

For any set $\mathbb{T} \subseteq U(d)$, we define $\mathbb{T}-$twirling of a channel $\mathcal{L}$ to be

$$\mathcal{L}^{\mathbb{T}} = \frac{1}{|\mathbb{T}|}\sum_{T\in\mathbb{T}} T\mathcal{L}T^\dagger \tag{A7}$$

The procedure in [24] deals with some specific error scenarios, in order to avoid the difficulties from general model, we need to add some necessary and reasonable assumption for the error model.

**Definition 1.** (GTM noise). A noise model is time-stationary if the noisy implementation $\tilde{\mathcal{U}}(t)$ of a gate $\mathcal{U}$ at time $t$ is a linear map that is independent of $t$ and if state preparations and measurements are respectively described by fixed operators and POVMs. A noise model for the Pauli group is called $GTM$ (gate-independent, time-stationary, Markovian) if it is time-stationary and there exists a completely positive trace-preserving map $\Lambda$ such that $\tilde{\mathcal{P}} = \mathcal{P}\Lambda$ for all $\mathcal{P} \in \mathbf{P}^n$.

**Definition 2.** (Weak, stable). A noise map $\mathcal{L}$ is $c-weak$ is the Pauli twirl $\mathcal{L}^{\mathbf{P}^n}$ is close to the identity channel in the operator norm. A SPAM parameter $A$ is called $c-stable$ if $A \geq 1-c$.

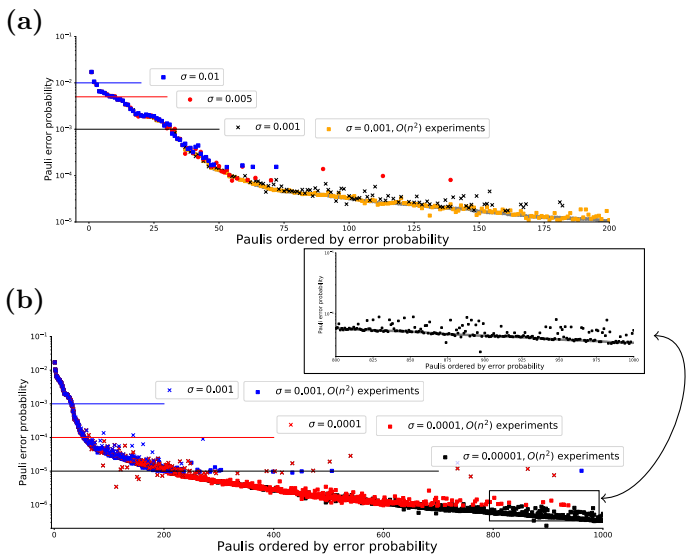



FIG. 3. **(a)** shows the additional recovery possible if protocol described in appendix B is used. Here we increase the number of experiments by varying each set of qubits over each of the 5 previously identified local stabiliser groups. This requires $O(n^2)$ experiments. On a practical level this is only required when many very low weight Paulis need to also be recovered, and will of course require many more measurements to obtain the fidelities with increasingly small amounts of noise. **(b)** shows the recover of 1000 different Paulis with error rates as low as $10^{-7}$ with high relative accuracy, even with a noise of $\sigma^2 = 10^{-5}$ applied to the fidelity results. Inset a more detailed look at the recovery in the regime between $10^{-6}$ and $10^{-8}$. with noise levels of $\sigma^2 = 10^{-5}$

## Appendix B: Experiments in the regime $\frac{1}{4} < \delta < \frac{1}{2}$

While the experimental protocol presented in the main text is likely to be all that is required in most practical regimes, if the number of Paulis to be recovered is high, then a slight modification might be needed. Unlike the situation where one is using completely random stabiliser groups, the local stabiliser protocol can fail when trying to reconstruct many low-error Paulis that differ only in one or two Paulis, in such a way that they cannot be separated by the local stabilisers. This might occur, for instance, in the regime where $\delta > 0.25$. In such circumstances, one can cycle each distinct set of two qubit pairs through the 5 stabiliser groups identified, and then generate the offset buckets for each of them. The number of experiments that need to be performed are the original experiment (1), then a further 4 for each qubit pair (4), times the number of qubit pairs $(n/2)$, times the number of experiments needed to generate the offsets on the remaining $n-2$ qubits $(4(n-2))$, for a total of $1 + 8n(n-2) = O(n^2)$ total experiments. The eigenvalues gathered this way allow the creation of $n/2$ properly offset sub-sampling matrices of $2^{n+2}$ buckets each containing $2^{n-2}$ Paulis. This appears to be sufficient to exactly recreate the global probabilities upto $\delta = 0.5$.

Figure 3 illustrates the extra recovery power available.

## Appendix C: Practical experimental and algorithm implementation - overview

In appendix E we prove theorem 1, giving recoverability guarantees, based on the assumptions contain in assumption 1. The full details of the components of the decoder required to support the recovery guarantee are contained in appendix D.

Here we describe in more detail the intuition behind the algorithm, the experiments prescribed and a simplified, practical extraction algorithm. [**?** ] contains code and examples showing how the following algorithm can be used to recreate the figures in this paper.

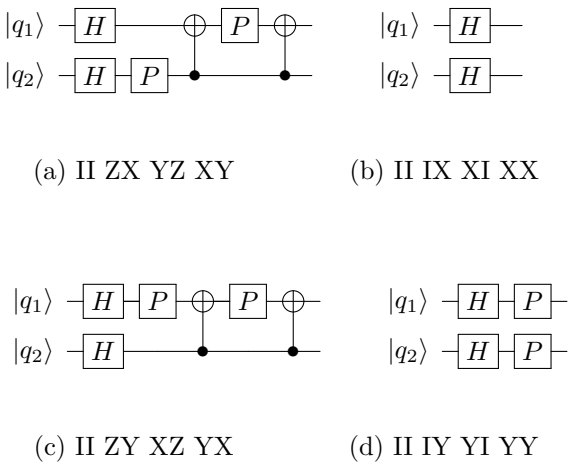
### Determining a suitable value for $C$

Our proofs relating to the recovery of $s$-sparse Pauli errors require an assumption that each element in the support set $\mathcal{K}$ is chosen independently and uniformly at random from $[N]$. At first glance it may appear that this is not likely to be the case in a quantum device as the Pauli errors are likely to cluster around low-weight Pauli errors rather than be uniformly distributed over the $4^n$ different possible Pauli errors. However where we choose random $n$-qubit stabilisers (global stabiliser groups) as the basis for sampling the Pauli eigenvalues, this effectively randomizes the "bin" into which we consign any specific Pauli error rate allowing us to satisfy the uniformly random distribution requirement.

Given this we can continue to use the "balls-and-bins" model utilized in [Lemma X]. We can use this insight, together with our ability to simultaneously sample $2^n$ commuting eigenvalues to determine practical values for $C$ given our bin size of $B = 2^n$.

Since the sparsity $s \ll N$, we have with high probability that the the maximum number of Paulis (balls) in one bin will be $\frac{1}{1-\delta}$, which in the sparsity regime of interest will be $< 2$. Assuming we have an $s-$sparse distribution, with $B := 2^n$ being the number of bins sampled. Then, the sparsity coefficient $\eta$, given by $B = \eta s$, is at least 1. This means that we only require $C$, being the number subsampling groups to be 2, to recover all of the edges in $O(s)$ iterations with probability at least $1 - O(1/s)$ (see [Insert Ref - was Appendix B of Li]). It can then be seen that as each experimental run recovers $2^n$ eigenvalues, we need to perform one experimental run for each bin plus one for each of the offsets of the bin times the number of subsampling groups. Giving the minimum number of experimental runs as 2(2n+1). As we discuss later by increasing the number of offsets we can increase the recovery guarantees, but at the cost of sampling more eigenvalues (although still scaling with $n$).

In the case where our device is too large for $\eta \geq 1$, for instance where we have had to marginalise over the measurements as $\log(B) \geq 30$, then we can increase our effective $C$ by marginalising over randomly chosen qubits and creating our sub-sampling matrices from such randomly chosen sub-samples of the measurement outcomes. This will allow us to retain the recovery guarantees without increasing the number of experiments on the device, although with increased computational cost in setting up and performing the peeling decoder.




(a) II ZX YZ XY (b) II IX XI XX

(c) II ZY XZ YX (d) II IY YI YY

FIG. 4. Some example sub-circuits required to perform the transform into the local stabiliser group appearing in Figure 1. The inverse gate will be of a similar form, but will have the relevant Paulis compiled into it, in order to undo the Pauli twirl. Note in (a) and (c) the order the stabilisers appear in the figure are important and different from the more natural ordering appearing in the body of the papers. The order can be changed by an appropriate circuit (to exactly match the text) but will require an extra two-qubit gate. Otherwise, care just needs to be taken in the operation of the decoder (and identification of the corresponding bucket). Finally the order of time is from left to right, i.e. left-most gates are applied first.

### Using local circuits

Our numerical simulations, based on available realistic data, indicate that randomly chosen global stabilisers are not in fact necessary to distribute the Pauli errors widely enough to allow recovery. It appears that local stabiliser groups suffice. This allows us to dramatically reduce circuit complexity while keeping the number of experiments required to a minimum. Figure 1(c) details the local two-qubit stabiliser groups that can be selected to perform an extraction experiment that is viable on most current devices. In fig. 4 we show the local Clifford circuits that can be used (with their corresponding inverse) at the beginning (end) of the measurement circuits to create these local stabiliser groups. In [24] it was shown that using such circuits we can estimate $2^n$ Pauli fidelities, with rel-

ative precision $\epsilon$, using $O(\epsilon^2 n)$ measurements.

Having chosen the series of stabilisers to measure, together with the circuits of offsets we will have all the relevant fidelities required to use the noisy peeling decoder.

In algorithm 1 we show how to operate the decoder on a practical level, assuming that there are a large number of Paulis sitting below the level of interest (i.e. with an error rate $\ll \epsilon$). To understand how it works one should note that there are two main components to dealing with the noise. The first is when deciding if the 'bucket' is zero, i.e. the only values in the bucket and its offsets are noise. We initially start willing to assume this is the case and slowly become less willing to accept that the bucket is really zero as we start to try and recover smaller and smaller error rates. This means that initially the decoder will concentrate only on buckets that have relatively large Pauli errors in them and will be less likely to mistake noise as indicative of an error.

For instance, assuming a reconstruction error normally distributed with a standard deviation of 0.01, then if a bucket contained $2^{14}$ 0-error Paulis with such noise, we would expect the mean of such a bucket to be centered around 0, with a standard deviation of $\sqrt{(0.01^2/2^{14})} \approx 7.8 \times 10^{-5}$. Therefore allowing 3 standard deviations we would expect the square of the noise in the bucket to be less that $\approx 5.5 \times 10^{-8}$. By ignoring buckets with a squared value less than $5.5 \times 10^{-8}$ and slowly decreasing this number to, say, $5.5 \times 10^{-10}$ one can ensure that higher error rate Paulis are first recovered, before exploring possibly empty buckets for low error rate Paulis. The second component is the willingness to accept that there is only one value in the bucket offsets. This time we start with a strict check, we only accept a Pauli if the noise is below a threshold and slowly relax this as we aim to recover Paulis that happen to have increasing amounts of noise in the buckets with them.

**Algorithm 1** Noisy Peeling Decoder

**Require:** M $\leftarrow$ Paulis in groups ($C$ sets), ▷ Figure 1
**Require:** $\lambda_{\Psi_{l,c}}$ for $l \in M_c$ $c \in [C]$, ▷ Eigenvalues
**Require:** $\lambda_{(\Psi_{l,c} \oplus b)}$ for $l \in M_c$ $c \in [C]$,$b \in 2^{[2n]}$ ▷ Offsets

$Z_s \leftarrow$ initial zero sensitivity
$S_s \leftarrow$ initial singleton sensitivity
$\mathcal{P} \leftarrow$ initialise empty list of Paulis + errors
▷ Set up quasi probability 'buckets'.
**for** c=1, ..., $C$ **do**
  $\tilde{p}_{j,c} \leftarrow \frac{1}{2^n} \sum_{l \in [2^n]} \lambda_{\Psi_{l,c}} (-1)^{\langle j,l \rangle}, j \in \mathbb{F}_2^n$ ▷ Equation (3)
  $\tilde{p}_{j,c,b} \leftarrow \frac{1}{2^n} \sum_{l \in [2^n]} \lambda_{(\Psi_{l,c} \oplus b)} (-1)^{\langle j,l \rangle}, j \in \mathbb{F}_2^n, b \in 2^{[2n]}$
**end for**
▷ Populate $\mathcal{P}$ with singletons.
**for** _ = 1, 2, ..., arbitrary **do**
  **for** c=1, ..., $C$ **do**
    **for** $\tilde{p} \in [\tilde{p}_{j,c}, \tilde{p}_{j,c,b}], j \in \mathbb{F}_2^n, b \in 2^{[2n]}$ **do**
      **if** not IsCloseToZero($\tilde{p}, Z_s$) **then**
        **if** IsSingleton($\tilde{p}, S_s$) **then**
          (P,E) $\leftarrow$ SingletonPauliAndSize($\tilde{p}$)
          $\mathcal{P} \leftarrow$(P,E)
          ▷ Remove from other sets
          PeelBack(M$_{[C]/c}$,(P,E))
        **end if**
      **end if**
      **if** $\sum \mathcal{P} \approx 1$ **then**
        **return** $\mathcal{P}$ ▷ Success!
      **end if**
    **end for**
  **end for**
  **if** no new Paulis added since previous iteration **then**
    $Z_s \leftarrow Z_s$ - Decrease willingness to assume zero
    $S_s \leftarrow S_s$ + Increase willingness to accept singleton
  **end if**
**end for**
**return** Incomplete $\mathcal{P}$ ▷ !Success

## Appendix D: Recovery Algorithm

In this section, we propose in detail a hashing-based subsampling recovery algorithm. Consider a Walsh-Hadamard transformation among $n$-qubit Pauli eigenvalues and Pauli error rates like (1). In order to recover a set of sparse Pauli error rates with noisy eigenvalues, it

is necessary to consider a noisy variation

$$\widehat{\lambda}_k = \sum_{m\in\mathbb{F}_2^{2n}} (-1)^{\langle k,m\rangle} p_m + w_k, \quad k \in \mathbb{F}_2^{2n}. \tag{D1}$$

Note that we employ $w_k$ to indicate the sampling errors of the eigenvalues, and for simplicity we are assuming that they are all independent Gaussian random variables with distribution $\mathcal{N}(0,\xi^2)$. The proposed algorithm follows the SPRIGHT framework of Ref. [27]. It first samples Pauli eigenvalues and forms several groups of bins. The algorithm implements the Peeling Decoder algorithm 3 to gain information from the sampled bins.

To subsample bins from noisy eigenvalues, this algorithm employs $C$ *subsampling groups*, each of which contains a binary matrix $\mathbf{M}_c \in \mathbb{F}_2^{2n\times 2b}$ and a set of $P$ offsets. These offsets provide redundancy designed make the recovery algorithm robust to such limited sampling noise. Before constructing explicit algorithms, we shall introduce a method for choosing offsets by using good error correcting codes [27].

**Definition 3.** *Let $P = P_1 + P_2$ with $P_i = O(n)$ for $i = 1, 2$. We choose $P_1$ random offsets $\mathbf{d}_t$ for $t = 0, \cdots, P_1-1$ chosen independently and uniformly at random over $\mathbb{F}_2^n$, and $P_2$ coded offsets $\mathbf{d}_t$ for $t = P_1, \cdots, P-1$ such that the offset matrix $\mathbf{G} = [\cdots; \mathbf{d}_t; \cdots;] \in \mathbb{F}_2^{P_2\times 2n}$ constitutes a generator matrix of a linear code with parameters $[P_2, 2n, \beta P_2]$ with $\beta > \mathbb{P}$, where $\mathbb{P}$ is the upper bound of the probability that the sample error will change the sign of a single-ton bin.*

In what follows, it will be convenient to define a $2n\times 2n$ matrix $J_n$ given by

$$J_n = \mathbf{I}_n \otimes X = \begin{pmatrix} \left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right) & & \\ & \ddots & \\ & & \left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right) \end{pmatrix}, \tag{D2}$$

where all unspecified entries are 0. This is the symplectic form that controls the commutivity relations in the Pauli group. That is, if $p, q \in \mathbb{F}_2^{2n}$ correspond to Paulis $P$ and $Q$ respectively, then $\langle p, q\rangle = p^T J_n q$ obeys $PQP = (-1)^{\langle p,q\rangle} Q$. (Note that most authors define $J_n = X \otimes \mathbf{I}_n$, but we have adopted this alternative convention to help illustrate our construction of the hash functions $\mathbf{M}_c^T m = j$.)

**Algorithm 2** Subsampling and WHT

```
Input : Offsets d_{c;t}, d_{c;t} for observation index t ∈ [P] and subsampling index c ∈ [C];
Input : Subsampling matrices M_c ∈ F_2^{2n×2b} for some b > 0 and c ∈ [C].
Modify : M'_c ← J_n M_c J_b ∀ c ∈ [C].
for all c ∈ [C], t ∈ [P], and ℓ ∈ F_2^{2b} do
    k ← M'_c ℓ + d_{c;t}
    Estimate : λ̂_k
end for
B ← 2^{2b}
for all c ∈ [C] and t ∈ [P] do
    U_{c;t}[j] ← (1/B) Σ_{ℓ∈F_2^{2b}} (-1)^{⟨j,ℓ⟩} λ̂_{M'_c ℓ + d_{c;t}}
    Output U_{c;t}[j]
end for
```

The indices on each array are considered to be modulo their respective dimension, and each element of the summation $\mathbf{M}'_c\ell + \mathbf{d}_{c;t}$ is calculated in the field $\mathbb{F}_2$. The algorithm calculates bin coefficients using the corresponding binary matrices and by taking sum over the whole field $\mathbb{F}_2^{2b}$. After this subsampling process, each subsampling group contains $P$ sets of $B = 2^{2b}$ bins.

**Lemma 1 (Basic Observation Model).** *The $B$-point WHT subsampled bin coefficients with index $j \in \mathbb{F}_2^{2b}$ can be written as:*

$$U_{c;t}[j] = \sum_{m:\, \mathbf{M}_c^T m = j} p_m (-1)^{\langle \mathbf{d}_{c;t}, m\rangle} + W_{c;t}[j], \forall t \in [P]. \tag{D3}$$

*Moreover the sample error is as follows*

$$W_{c;t}[j] = \sum_{m:\, \mathbf{M}_c^T m = j} W_m (-1)^{\langle \mathbf{d}_{c;t}, m\rangle},$$

*where $W_m$ is the WHT coefficient of noise samples $w_k$.*

*Proof.* Denote $p_m + W_m$ by $\tilde{p}_m$, so the noisy Pauli eigenvalues can be transformed to

$$\widehat{\lambda}_k = \sum_{m\in\mathbb{F}_2^{2n}} (-1)^{\langle k,m\rangle} \tilde{p}_m, \ \forall\, k \in \mathbb{F}_2^{2n}.$$

Therefore, from Algorithm 2, we can compute that for all $t \in [P]$

$$U_{c;t}[j] = \frac{1}{B} \sum_{\ell\in\mathbb{F}_2^{2b}} \sum_{m\in\mathbb{F}_2^{2n}} (-1)^{\langle m, \mathbf{M}'_c\ell\rangle} (-1)^{\langle j,\ell\rangle} (-1)^{\langle m, \mathbf{d}_{c;t}\rangle} \tilde{p}_m.$$

The key observation is

$$\langle m, \mathbf{M}'\ell\rangle = m^T J_n \mathbf{M}'\ell$$

$$
\begin{aligned}
&= m^T J_n \cdot J_n \mathbf{M} J_b \ell \\
&= (\mathbf{M}^T m)^T J_b \ell \\
&= \langle \mathbf{M}^T m, \ell \rangle,
\end{aligned}
$$

where the second equation comes from `Modify` part in Algorithm 2, and the third is due to the following property of $J$

$$
J_n \cdot J_n = \mathbf{I}_{2n} \ \forall\, n \in \mathbb{Z}^*. \tag{D4}
$$

Thus the bins can be simplified as follows,

$$
\begin{aligned}
U_{c;t}[j] &= \frac{1}{B} \sum_{m \in \mathbb{F}_2^{2n}} \sum_{\ell \in \mathbb{F}_2^{2b}} (-1)^{\langle \mathbf{M}_c^T m + j, \ell \rangle} (-1)^{\langle m, \mathbf{d}_{c;t} \rangle} \tilde{p}_m \\
&= \sum_{m:\, \mathbf{M}_c^T m = j} \tilde{p}_m (-1)^{\langle m, \mathbf{d}_{c;t} \rangle} \\
&= \sum_{m:\, \mathbf{M}_c^T m = j} p_m (-1)^{\langle \mathbf{d}_{c;t}, m \rangle} + W_{c;t}[j],
\end{aligned}
$$

where $W_{c,t}[j]$ is defined in the lemma. □

It is clear from the calculation in algorithm 2, the noisy part in (D3) remains a Gaussian distribution $\mathcal{N}(0, \nu^2)$ where $\nu^2 = \frac{\xi^2}{B}$. Moreover, we can combine each $U_{c;t}[j]$ for different $t \in [P]$, and get a vector

$$
\mathbf{U}_c[j] := [U_{c;0}[j], \cdots, U_{c;P-1}[j]]^T,
$$

and an analogous vectorization can be implemented on offsets

$$
\mathbf{D}_c := [\mathbf{d}_{c;0} \cdots \mathbf{d}_{c;P-1}].
$$

Therefore, Lemma 1 can be rewritten as follows.

**Lemma 2 (Bin Observation Model).** *The $B$-point WHT subsampled bin with index $j \in \mathbb{F}^{2b}$ in the $c$-th subsampling group is*

$$
\mathbf{U}_c[j] = \sum_{m:\, \mathbf{M}_c^T m = j} p_m (-1)^{\langle \mathbf{D}_c, m \rangle} + \mathbf{W}_c[j], \tag{D5}
$$

*where $\mathbf{W}_c[j] = \sum_{m:\, \mathbf{M}_c^T m = j} W_m (-1)^{\langle \mathbf{D}_c, m \rangle}$ and $W_m$ is the WHT coefficient of noise samples $w_k$.*

*Proof.* This is a variation of Lemma 1. □

After subsampling and calculating bins, it's straightforward to design a protocol to extract information from these bins. The idea is to construct a bipartite graph $G$ with $s$ left nodes representing nonzero Pauli error rates and $BC$ right nodes representing bin vectors $\mathbf{U}_c[j]$.

We draw an edge from each left node (a nonzero Pauli error rate) to every right node that contains that Pauli. Each Pauli error rate will occur exactly once in each subsampling group, the degree of the left nodes is therefore C. We can use the resulting degrees of the right hand nodes to partition them into three types. We call a bin with only one nonzero Pauli error rate a *single-ton*, and similarly there are *zero-ton* and *multi-ton* bins that contain zero or more than one Pauli error rate respectively. Shortly we will describe in detail a method to detect which type of bin a particular node has been partitioned into. After invoking such a bin detector, the peeling decoder can be designed to peel out the detected single-ton Pauli error rates by subtracting them from every multi-ton bin in which they appear, removing the associated edge from the graph. This will reduce the degree of that right-hand node, potentially turning it from a multi-ton bin into a singleton bin. For the range of parameters that we have chosen and the assumptions outlined above, iterating this decoder to discover new single-tons and reduce multi-tons will converge to reduce the graph to zero-ton and single-ton bins with high probability.

One subtlety to applying the peeling decoder to this graph is that the graph might have cycles. Peeling on a graph with cycles will in general lead to dependencies in the random variables, which complicates the analysis. However, as we show below in lemma 5, large local neighborhoods of the peeling graph look locally tree-like with high probability, therefore we can peel for a large number of steps before encountering a cycle. With the correct choice of parameters, the tree-like neighborhood can be made large enough throughout the graph to ensure convergence of the peeling decoder.

In the algorithm below, we apply an array $\mathbf{T}$ that indicates the variance of the propagated noise part, $W_{c;t}[j]$, in each bin. These numbers help track the bin detector error propagation from the calculation in line 10 of Algorithm 3. The equation in line 9 of that algorithm describes how to update $\mathbf{T}$. Lemma 7 shows the need and utility of this parameter.

The peeling decoder algorithm is based on a subroutine that we call *Bin Detector* (it is set out in algorithm 4). We will denote it by $\mathrm{BD}(\mathbf{U}_c, \mathbf{D}_c, T)$. The subroutine, BD, will take a bin, the offsets chosen and a parameter $T$ as inputs, and it will output an estimate for the type of the bin $\widehat{\mathfrak{B}}$, and if the bin is a single-ton it also returns the estimated index $\widehat{m}$ and Pauli error rate $\widehat{p}_{\widehat{m}}$.

**Algorithm 3** Peeling Decoder

**Input** : observation vectors $\mathbf{U}_c[j]$, offsets $\mathbf{D}_c$ and array $\mathbf{T}_c[j]$ initialized by 1 for $j \in \mathbb{F}_2^{2b}$, $c \in [C]$;
**Input** : the number of peeling iterations $I$;
**for** $i \in [I]$ **do**
  **for all** $c \in [C]$ and $j \in \mathbb{F}_2^{2b}$ **do**
    $(\widehat{\mathfrak{B}}, \widehat{m}, \widehat{p}_{\widehat{m}}) \leftarrow \mathrm{BD}(\mathbf{U}_c[j], \mathbf{D}_c, \mathbf{T}_c[j])$
    **if** $\widehat{\mathfrak{B}} =$ single-ton **then**
      **for all** $c' \in [C]$ and $c' \neq c$ **do**
        `Locate bin index` $j_{c'} \leftarrow \mathbf{M}_{c'}^T \widehat{m}$
        $\mathbf{T}_{c'}[j_{c'}] \leftarrow \mathbf{T}_{c'}[j_{c'}] + \frac{\mathbf{T}_c[j]}{P_1} - \frac{(P_1+1)B}{P_1 N}$
        $\mathbf{U}_{c'}[j_{c'}] \leftarrow \mathbf{U}_{c'}[j_{c'}] - \widehat{p}_{\widehat{m}}(-1)^{\langle \mathbf{D}_{c'}, \widehat{m} \rangle}$
      **end for**
    **else if** $\widehat{\mathfrak{B}} \neq$ single-ton **then**
      continue to next $j$.
    **end if**
  **end for**
**end for**

According to the sparsity assumption, this peeling process will succeed with high probability. Intuitively we can see this from our ability to choose subsampling matrices $\mathbf{M}$ in such a way that we can find bins that typically contain only zero or one nonzero Pauli error rate. In [27] the authors provide a proof that if the bin detector BD always returns an exactly correct answer, then the oracle-based peeling decoder has a failure probability that vanishes with the sparsity. So it suffices to propose a suitable design for the bin detector and a corresponding recovery guarantee.

In designing such a bin detector we will need to estimate the index of the relevant Pauli in the bin. For index estimation in the setting when there is noise we will need to make our estimation robust. One approach is to use some repetition of detection and a majority voting. A better approach is to use some form of error correcting code. Lemma [?] confirms that offsets chosen in accordance with Definition 3 can be used to estimate the indices. In what follows, we will use the following definition of a sign function,

$$\mathsf{sgn}\,[x] = \begin{cases} 0 & \text{if } x \geq 0, \\ 1 & \text{if } x < 0. \end{cases} \tag{D6}$$

With this definition, we have the following lemma.

**Lemma 3.** *Given a single-ton bin $(m, p_m)$ observed with noise*

$$U = p_m(-1)^{\langle \mathbf{d}, m \rangle} + W, \tag{D7}$$

*and supposing that the variance in each row (offset, $\mathbf{d}$) of the bin is equal to $T\nu^2$, then the sign of each observation satisfies*

$$\mathsf{sgn}\,[U] = \langle \mathbf{d}, m \rangle \oplus Z, \tag{D8}$$

*where $Z$ is a Bernoulli random variable with probability upper bounded as $\mathbb{P}_m = \sqrt{\frac{T\nu^2}{2\pi p_m^2}} \mathrm{e}^{-\frac{p_m^2}{2T\nu^2}}$.*

*Proof.* The first term in (D8) follows trivially from the sign of the power of minus one in (D7) and the fact that $p_m$, being a probability, is always positive. The second term, $Z$, will be 1 if and only if $|W|$ is larger than $p_m$ so that it can change the sign generated by the first term of (D7). Therefore, $Z$ is Bernoulli distributed with a probability we can bound as follows:

$$\begin{aligned} \Pr(Z=1) &= \frac{1}{2}\Pr(|W| > p_m) = \Pr(W > p_m) \\ &\leq \sqrt{\frac{T\nu^2}{2\pi p_m^2}} \mathrm{e}^{-\frac{p_m^2}{2T\nu^2}} = \mathbb{P}_m, \end{aligned} \tag{D9}$$

where $T$ is the number extracted from the array $\mathbf{T}$ in Algorithm 3. The last inequality follows from a tail bound of a normal distribution and our assumption on the variance, for details see [30]. □

**Remark 1.** If we assume that the maximum degree of right nodes in the bipartite graph $G$ is not larger than $P_1$, $\mathbb{P}_m < \frac{1}{2}$ is satisfied for all $m \in \mathbb{F}_2^{2n}$ (using **A4** in Assumption 1 and Lemma 9). We will bound the probability that this right-hand degree assumption is not true in Lemma 8.

In what follows, we will ignore the subscripts $c$ and indices $j$ of the bins when it will not lead to any misunderstanding.

Now Lemma 3 can be used to identify $\widehat{m}$, the index of the Pauli error rate in a single-ton bin. Given the offsets chosen in Definition 3 and recalling Lemma 1, we have the following equation from the code generator $\mathbf{G}$ for the signs of every element in a bin,

$$\begin{bmatrix} \mathsf{sgn}\,[U_{P_1}] \\ \vdots \\ \mathsf{sgn}\,[U_{P-1}] \end{bmatrix} = \langle \mathbf{G}, m \rangle \oplus \begin{bmatrix} Z_{P_1} \\ \vdots \\ Z_{P-1} \end{bmatrix}. \tag{D10}$$

Since the bit length of index $m$ is $2n$, we can choose the number $P_2$ as follows. We choose any linear code with rate $R$ and distance $d$, and a decoder that can decode up to at least a minimum distance $\beta 2n/R$ for parameters $\beta, R = \Theta(1)$. Obviously this requires $d \geq \beta 2n/R$. The additional constraint on $\beta$ is that $\beta$ is larger than the probability $\mathbb{P}$ of any of the Bernoulli random variables $Z_i$ to be 1. Then we can choose $P_2 = 2n/R$. That is, we are looking for a classical linear code with parameters $[2n/R, 2n, d \geq \beta 2n/R]$. There are a number of pre-existing candidate codes that can be decoded up to a

constant fraction of the minimal distance in linear time in the length of the code exist that satisfy these stringent conditions. For example, expander codes [31] can be implemented to construct the code generator $\mathbf{G}$ and the parity check matrix $\mathbf{H}$, and the greedy linear-time decoder [31] can correct errors with weight up to $d/4$. The decoder of the corresponding code is required to retrieve the estimate $\hat{m}$. Since the manner of coding and decoding is flexible, here we only use `Decode` to indicate the decoder.

$$\widehat{m} = \texttt{Decode}\left(\begin{bmatrix} \mathsf{sgn}\left[U_{P_1}\right] \\ \vdots \\ \mathsf{sgn}\left[U_{P-1}\right] \end{bmatrix}\right). \tag{D11}$$

With this we can specify the modified algorithm to detect the bin $U$ with the offsets as in Definition 3 along with the corresponding number $T$.

We are now ready to give a precise specification of the bin detector algorithm.

**Algorithm 4** Bin Detector: $\mathrm{BD}(\mathbf{U}_c, \mathbf{D}_c, T)$

`Input`: bin $\mathbf{U}_c$, offsets $\mathbf{D}_c$ and the number $T$ to indicate error size;
`Parameter`: real numbers $\gamma_1, \gamma_2 \in (0,1)$;
**if** $\frac{1}{P_1}\sum_{t=0}^{P_1-1} U_{c;t}^2 \leq T(1+\gamma_1)\nu^2$ **then**
  $\widehat{\mathfrak{B}} \leftarrow$ `zero-ton`
  Return $(\widehat{\mathfrak{B}}, \textit{nil}, \textit{nil})$ ▷ *zero-ton verification*
**end if**
$\widehat{m} \leftarrow \mathrm{Decode}([\mathsf{sgn}\left[U_{P_1}\right], \cdots, \mathsf{sgn}\left[U_{P-1}\right]]^T)$
$\widehat{p}_{\widehat{m}} \leftarrow \frac{1}{P_1}\sum_{t=0}^{P_1-1}(-1)^{\langle \mathbf{d}_{c;t}, \widehat{m}\rangle} U_{c;t}$ ▷ *single-ton search*
**if** $\frac{1}{P_1}\sum_{t=0}^{P_1-1}(U_{c;t} - (-1)^{\langle \mathbf{d}_{c;t}, \widehat{m}\rangle}\widehat{p}_{\widehat{m}})^2 \leq T(1+\gamma_2)\nu^2$ **then**
  $\widehat{\mathfrak{B}} \leftarrow$ `single-ton`
  Return $(\widehat{\mathfrak{B}}, \widehat{m}, \widehat{p}_{\widehat{m}})$ ▷ *single-ton verification*
**else**
  $\widehat{\mathfrak{B}} \leftarrow$ `multi-ton`
  Return $(\widehat{\mathfrak{B}}$,*nil, nil*$)$
**end if**

## Appendix E: Proof of Main Theorem

The following discussion about each kind of failure along with the bound of the oracle-based peeling decoder processing in [27] shows an exponentially decaying bound of the failure probability.

We now repeat the statement of the main theorem.

**Theorem 1′.** *Let Assumption 1 hold for the target Pauli eigenvalues $\widehat{\lambda}$ and the corresponding Pauli error rates $\mathbf{p}$. Then for any sparsity regime $s = O(4^{n\delta})$ with $0 < \delta < 1$, the above algorithm computes the $s$-sparse Pauli error rates $\widehat{\mathbf{p}}$ with a noise bound $||\widehat{\mathbf{p}} - \mathbf{p}||_\infty \leq \sqrt{4\nu^2}$, where by retaining a sample complexity that scales only as $O(sn)$ and with a computation complexity that scales as $O(sn^2)$ the algorithm achieves a vanishing failure probability $\mathbb{P}_F$ for sufficiently large $n$.*

*Proof.* Firstly we consider the stipulated sample and computational complexities. From Theorem 2 in [27], it's shown that the oracle-based peeling decoder succeeds with probability $1 - O(1/s)$ for a sparse set as long as $C = O(1)$ and $B = O(s)$. Therefore, to prepare these bins, the sample complexity of this system is $M = BPC = O(sP)$.

To construct the bins and the corresponding graph, the computational complexity can be calculated by the complexity from the construction algorithm. Note there are $P$ offset coefficients $\mathbf{d}$ and each $\mathbf{U}_{c,t}[j]$ comes from the sum of $B$ samples in Algorithm 2. To construct the total set $\{\mathbf{U}_{c,t}[j]\}_{C,P,B}$, the we can use a fast WHT (which has complexity $O(B\log B)$ to calculate a $B$-point WHT) for each offset. Therefore, the computational complexity for this part is

$$Z_1 = O(PB\log B) = O(Psn).$$

The second part of computational complexity comes from the computation of Algorithm 3. Each step in the bin detector checks the character of the bin with $O(P)$ calculations, and there are $O(B)$ iterations. Accordingly, the complexity is

$$Z_2 = O(PB) = O(Ps).$$

Therefore, the total computational complexity is $Z = Z_1 + Z_2 = O(Psn)$.

Utilising the law of total probability we can bound the failure rate of the entire algorithm as

$$\begin{aligned}\mathbb{P}_F \leq & \Pr\left(\text{Peeling decoder fails}\middle| E_{\text{bin}}^c\right) \\ & + \Pr\left(E_{\text{bin}}|D,H\right) + \Pr\left(D^c\right) + \Pr\left(H^c\right),\end{aligned} \tag{E1}$$

where $E_{\text{bin}}$ denotes the event that any invocation of the bin detector (in the execution of Algorithm 3) returned one or more of the following: (a) an incorrect identification of the type of bin; (b) wrong indices for a detected single-ton, or (c) a mis-estimate of the Pauli error rates of a detected single-ton by more than $\sqrt{4\nu^2}$. $D$ denotes the event the maximum degree of right nodes in $G$ is less or equal than $P_1$. Let $H$ be the event that all the peeling routes are in the procedure are cycle-free. The subscript $c$ denotes the complement of the event, e.g. $E_{\text{bin}}^c$, denotes that no bin detection error occured in the entire execution of Algorithm 3.

The first term in (E1) is the chance that the oracle-based peeling decoder fails, even though the bin decoder is always correct. This probability scales as $O(1/s)$ (Proposition 4 in [27]).

To bound the second term, it will be more convenient to consider the probability that every invocation of a bin

detector works correctly given $D$ and $H$. Let $M$ denote the number of times the peeling decoder calls the bin detector subroutine. This probability can be expressed as follows,

$$\begin{aligned}&\Pr\left(\bigcap_{i=1}^{M} E_i^c|D,H\right)\\ =&\Pr\left(E_M^c|\bigcap_{i=1}^{M-1} E_i^c,D,H\right)\Pr\left(\bigcap_{i=1}^{M-1} E_i^c|D,H\right)\\ =&\Pr\left(E_M^c|\bigcap_{i=1}^{M-1} E_i^c,D,H\right)\cdots\Pr\left(E_1^c|D,H\right),\end{aligned}$$

where $E_i$ denotes the event that the $i$th call of bin detector returns a wrong answer. According to Lemma 7, the parameter $T$ will always correctly estimate the variance if all the earlier bin detectors work correctly. From Lemma 10, each term in the above equation will be lower bounded by

$$\Pr\left(E^c|D,V,H\right)\geq 1-\mathrm{e}^{-O(P_1)},$$

where $V$ here just indicates that all the previous bin detectors work correctly. So we have

$$\begin{aligned}\Pr\left(\bigcap_{i=1}^{M} E_i^c|D,H\right)&\geq\left(1-\mathrm{e}^{-O(P_1)}\right)^M\\ &\geq 1-M\mathrm{e}^{-O(P_1)}.\end{aligned}$$

Moreover, since $M$, the number of times the bin detector routine is called, is at most $BCs$, the upper bound of the second term is

$$\Pr\left(E_{\text{bin}}|D,H\right)\leq BCs\mathrm{e}^{-O(n)}\leq\mathrm{e}^{-O(n)}.$$

Lemma 8 provides that the third term in (E1) is also exponentially decaying with $P_1$:

$$\Pr\left(D^c\right)\leq\mathrm{e}^{-O(P_1)}\leq\mathrm{e}^{-O(n)},$$

where the last inequality comes from the definition of $P_1$ from Definition 3. Similarly Lemma 5 and Remark 2 provide the magnitude of probability $H^c$:

$$\Pr\left(H^c\right)\leq O\left(\frac{\log^{\log\log(s)}s}{s}\right)\leq\mathrm{e}^{-O(n)},$$

Therefore, the total failure rate of our peeling decoder algorithm is vanishing as the number of qubits $n$. And from Definition 3, the total offset consists of random offset $P_1=\Theta(n)$ and the coding offset $P_2=\Theta(n)$, thus $P=\Theta(n)$ and complexities have been proved. □

## Appendix F: Tail bounds

In this section, we prove several statements bounding the failure probabilities of various events that can cause the bin detector outlined as algorithm 4 to fail.

One of the main lemmas that we will need is the following tail bound on Gaussian random variables.

**Lemma 4** (Tail bound [27, Lemma 11])**.** *Given $\boldsymbol{g},\boldsymbol{k}\in\mathbb{R}^N$ where $\boldsymbol{k}$ is an isotropic Gaussian random variable $\boldsymbol{k}\sim\mathcal{N}(0,\nu^2\mathbb{1}_N)$, then the following tail bound holds:*

$$\Pr\left(\frac{1}{N}\|\boldsymbol{g}+\boldsymbol{k}\|^2\geq\tau_1\right)\leq\mathrm{e}^{-\frac{N}{4}\left(\sqrt{2\tau_1/\nu^2-1}-\sqrt{1+2\theta_0}\right)^2} \tag{F1}$$

$$\Pr\left(\frac{1}{N}\|\boldsymbol{g}+\boldsymbol{k}\|^2\leq\tau_2\right)\leq\mathrm{e}^{-\frac{N}{4}\frac{\left(1+\theta_0-\tau_2/\nu^2\right)^2}{1+2\theta_0}}, \tag{F2}$$

*for $\tau_1,\tau_2$, and $\theta_0$ satisfying*

$$\tau_1\geq\nu^2(1+\theta_0),\ \tau_2\leq\nu^2(1+\theta_0),\ \theta_0=\frac{\|\boldsymbol{g}\|^2}{N\nu^2}.$$

Since we will use this Lemma 4 to get a failure bound, it is critical to show the sample errors within the different offsets for a particular bin are independent. However, given that bins are created as shown in line 10 in Algorithm 3, it is not immediately clear that the sample errors remain independent. To show this independence let us first extend the definition of the sample errors $W_{c;t}[j]$ to take into account the effect of the peeling decoder algorithm 3. Recall that for any particular bin we have P (being, P1+P2) offset bins.

**Definition 4.** *For a specific bin, regard $\mathbf{U}_c[j]$ as a vector of length $P$ as in Lemma 2. Denote the set of indices of the current contained non-zero Pauli error rates by $\mathcal{P}$. Denote the P1 sample errors, being the sample errors that occur in the offsets of that bin where the offset indices are $t\in[P_1]$, for a certain timestamp in the peeling decoder $W_{c;t}[j]$ as*

$$W_{c;t}[j]:=U_{c;t}[j]-\sum_{m\in\mathcal{P}}(-1)^{\langle\mathbf{d}_{c;t},m\rangle}p_m.$$

*And similarly denote the P2 sample errors, being those those that occur in the bins with offset indices $t\in[P_2]$, as*

$$W_{c;t+P_1}[j]:=U_{c;t+P_1}[j]-\sum_{m\in\mathcal{P}}(-1)^{\langle\mathbf{d}_{c;t+P_1},m\rangle}p_m.$$

*We can combine all of these sample errors to be $\mathbf{W}_c[j]$ following the manner of Lemma 2.*

In case of the discussion about the independence of errors and the variance evolution, we first introduce some results to rule out some intricate situations. To define this more rigorously, consider the

*directed neighborhood* $\mathcal{N}_e^l$ in the bipartite graph consists of nonzero Paulis (left nodes) and all the bins (right nodes). This $\mathcal{N}_e^{2l}$ with length $2l$ and an edge $e=(v,c)$ is an induced sub-graph containing all edges and nodes on paths $e_1, e_2 \cdots, e_{2l}$ from node $v$ where $e \neq e_1$.

Denote the event that for every edge in the bipartite graph, this sub-graph $\mathcal{N}_e^{2l}$ is cycle-free by $\mathcal{T}_l$. If $\mathcal{T}_l$ occurs all the first $l$ peeling iterations will progress independently and there is no initial error propagating to a single bin with more than once in the first $l$ iterations. It has been shown in [27] that with sufficiently large $s$ and $N$, the effective part of the bipartite graph behaves with a cycle-free manner.

**Lemma 5** (Ref. [27, Lemma 6]). *For any iteration $l$, the probability of the complement of $\mathcal{T}_l$ is bounded as*

$$\Pr\left(\mathcal{T}_l^c\right) \leq c_0 \cdot \frac{\log^l s}{s},$$

*for some constant $c_0$.*

**Remark 2.** Li *et al* [27] shows the probability $p_l$ to depict an arbitrary edge to be held after $l$ peelings given the sub-graph is tree-like or cycle-free is calculated in an iterative approach.

$$p_l = \begin{cases} 1 & l=0 \\ \left(1-\mathrm{e}^{-\frac{p_{l-1}}{\eta}}\right)^{C-1} & \text{otherwise} \end{cases}, \tag{F3}$$

where $\eta$ is the factor $\frac{B}{s}$ and $C$ is the number of subsampling groups.

If we take $C=6$ and $\eta=1$, which is a reasonable setting, the probability will decrease to the *machine epsilon* in only three iterations. And in general this $p_l$ vanishes exponentially with the exponent with of $l$. With the law of total probability, the probability of that there exist any edges after $l \sim O(\log\log(s))$ iterations, denoted by $h^c$, is

$$\Pr(h^c) = O\left(\frac{\log^{\log\log(s)} s}{s}\right). \tag{F4}$$

Therefore, the event that there exist bins getting peeled by some earlier bins in a cycle manner during the whole process happens with probability of the same magnitude of (F4), which is convergent to zero as $s$ is sufficiently large.

**Lemma 6.** *For an arbitrary timestamp in Algorithm 3, sample errors in each of the P1 sample errors for a particular bin $\mathbf{U}_c[j]$ remain independent of each other given the peeling route is cycle-free.*

*Proof.* For an initial bin subsampled from Algorithm 2, consider an arbitrary pair of offsets labeled by $t_1, t_2 \in [P_1]$ in the same bin $\mathbf{U}_c[j]$

$$W_{c;t_1}[j] = \sum_{m:\,\mathbf{M}_c^T m = j} W_m (-1)^{\langle \mathbf{d}_{c;t_1}, m\rangle},$$
$$W_{c;t_2}[j] = \sum_{m:\,\mathbf{M}_c^T m = j} W_m (-1)^{\langle \mathbf{d}_{c;t_2}, m\rangle}.$$

Since all the errors $W_m$ are i.i.d. Gaussian random variables $\mathcal{N}(0, \frac{\xi^2}{N})$, it is obvious that $\mathbb{E}(W_{c;t_1}[j]\cdot W_{c;t_2}[j]) = 0$. So they are independent given that the expected values of the samples errors are 0.

The peeling decoder in line 10 in Algorithm 3 causes errors in the estimate of $W_{c;t}[j]$ to propagate in the following manner

$$W_{c;t}[j] \leftarrow W_{c;t}[j] + (p_m - \widehat{p}_{\widehat{m}})(-1)^{\langle \mathbf{d}_{c;t}, \widehat{m}\rangle}.$$

We now proceed by induction. As discussed above the noise is initially independent, so the base case is satisfied. Now assume that the sample errors before peeling are independent of each other. Observing that the updated error still has mean zero, we can calculate the expected value of a product between an arbitrary pair of sample errors in the offsets of a bin to show independence as between the offset bins. For convenience, denote the updated error by $W_{c;t}[j]'$. Then we have

$$\begin{aligned}
&\mathbb{E}(W_{c;t_1}[j]' \cdot W_{c;t_2}[j]') \\
&= \mathbb{E}(W_{c;t_1}[j] \cdot W_{c;t_2}[j] + (p_m - \widehat{p}_{\widehat{m}})(-1)^{\langle \mathbf{d}_{c;t_2} + \mathbf{d}_{c;t_1}, \widehat{m}\rangle} \\
&+ W_{c;t_1}[j] \cdot (p_m - \widehat{p}_{\widehat{m}})(-1)^{\langle \mathbf{d}_{c;t_2}, \widehat{m}\rangle} \\
&+ W_{c;t_2}[j] \cdot (p_m - \widehat{p}_{\widehat{m}})(-1)^{\langle \mathbf{d}_{c;t_1}, \widehat{m}\rangle}) = 0.
\end{aligned}$$

Note the last two terms are expected to be zero since in each term the original error is independent from the additional one according to the condition that the peeling process is cycle-free. □

In Algorithm 3, we employ an array $\mathbf{T}$ to keep track of the variance of sample error for each bin. This array gets updated whenever the algorithm peels a bin using an estimated Pauli error rate. We now show that this does indeed correctly track the variance of the sample errors in the bins.

**Lemma 7.** *Suppose that at a given arbitrary timestamp in Algorithm 3 all the bin detector subroutines called earlier have correctly identified their bins. And the peeling route is cycle-free. Then for each bin and its corresponding offsets $\mathbf{U}_c[j]$, the sample error for that bin and each of its offsets $\mathbf{W}_c[j]$ have the same variance $T_c[j]\cdot\nu^2$.*

*Proof.* Since this statement is based on the premise that all of the earlier bin detector runs were accurate, we can assume that the index $\widehat{m} = m$ is correct. We will still write $\widehat{m}$ to distinguish these index estimates from the original index $m$. The idea is to calculate the variance after a peel by induction with the fact that for any two

random variables,

$$\mathrm{Var}\left(X+Y\right)=\mathrm{Var}\left(X\right)+\mathrm{Var}\left(Y\right)+2\mathrm{Cov}\left(X,Y\right). \quad \text{(F5)}$$

Assume that the statement in the lemma holds before a peeling. Then we need to show that if we subtract an estimated Pauli, $\widehat{p}_{\widehat{m}}$, from a bin in a different subsampling group that contains this Pauli, the statement is preserved as the updating of $T$. To do this we will work out the variance of the sample error in each term and the covariance between these sample errors. Armed with this we will be able to prove that the statement is preserved after peeling.

We now consider the peeling process causes error propagation for the error part of the bin $\mathbf{U}_c[j]$ as follows

$$W_{c;t}[j]\leftarrow W_{c;t}[j]+(p_m-\widehat{p}_{\widehat{m}})(-1)^{\langle \mathbf{d}_{c;t},\widehat{m}\rangle}. \quad \text{(F6)}$$

The variance of the first term is by induction $T_c[j]\nu^2$, while the variance of the second term is not so trivial. The estimated Pauli comes from the *single-ton search*, where all the first $P_1$ observations get summed after added a random sign. Since all the random signs of $W_{\widehat{m}}$ terms are annihilated before summation, and all the other error parts still remain random, the variance of this second term in (F6) is

$$\mathrm{Var}\left(p_m-\widehat{p}_{\widehat{m}}\right)=\frac{T_{c'}[j']\times\nu^2}{P_1}+\frac{(P_1-1)B\times\nu^2}{P_1N}. \quad \text{(F7)}$$

Because of the assumption that all the initial errors in different bins are independent and the condition that the whole peeling routes are cycle-free, it can be viewed obviously that these two term remains independent. Therefore, we have calculated variance as follows

$$\mathrm{Var}\left(W_{c;t}[j]\right)_{\text{after}}/\nu^2=T_c[j]+\frac{T_{c'}[j']}{P_1}+\frac{(P_1-1)B}{P_1N},$$

which is consistent with what we have claimed. □

In order to prove Theorem 1 and find a bound on the variances of the sample errors, it is necessary to find an upper bound on the parameter $T$, which needs to be analyzed for both the graph and the algorithm. Denote by $G$ the bipartite graph of which each right node represents a bin observation, each left node represents a nonzero Pauli error rate, and edges come from the hash function relation:

$$\mathbf{M}_c^T m=j. \quad \text{(F8)}$$

That is, a bin-observation-node $\mathbf{U}_c[j]$ is connected to error-rate-node $p_m$ if and only if it holds that $\mathbf{M}_c^T m=j$. A right node is a single-ton node if and only if it has a single edge connected to it. Every time we peel a left node (that is we identify a Pauli error) we remove the edges connecting it and the right nodes. Each peeling therefore decreases the degree of the right nodes.

The following two lemmas help us bound the integer array $\mathbf{T}$ as we peel along the graph $G$. We first bound the right degree of $G$.

**Lemma 8.** *The maximum degree of the right nodes in $G$ is less or equal than $\frac{P_1}{2}$ with probability $1-\mathrm{e}^{-O(n)}$.*

*Proof.* Put the right nodes in some sequential order, and define events $\{X_i\}_{i=1}^{BC}$ where $X_i$ denotes the $i$th node and is linked to more than $P_1/2$ left nodes. According to the bin observation model (D5), each bin connects with $\frac{N}{B}$ Pauli error rates (most of which will be zero), so the expected degree of a right node is $\frac{s}{B}$ where $s$ is the number of left nodes. Since the protocol chooses $B=O(s)$, the expected degree $e=\frac{s}{B}\sim O(1)$.

Note that by Assumption 1, the support set of the Pauli error rates is chosen randomly. Therefore, concentrating on a specific bin $i$, we can introduce a random variable $d_i$ that denotes the degree of bin-observation-node $i$ (a right node), and introduce the variable $d_{ij}$ which is 0 if the corresponding $j$th Pauli error rate is zero or if $p_j$ is not in the $i$th bin, and 1 otherwise. Then we have the relation

$$d_i=\sum_j d_{ij}, \quad \text{(F9)}$$

since this counts the support in the $i$th bin. These variables $d_{ij}$ are actually all Bernoulli variables and the only correlation among them comes from the constraint that there are exactly $s$ elements in the entire support. This constraint means that the $d_{ij}$ are negatively correlated, and so the probability of $d_{ij}=1$ can be upper bounded by considering the event that all the other Pauli rates linked with this bin are zero,

$$\Pr\left(d_{ij}=1\right)\leq\frac{sB}{N(B-1)}.$$

Now consider another set of i.i.d. Bernoulli variables $\{d'_{ij}\}_j$ each of which is 1 with probability $\frac{sB}{N(B-1)}$. We then have

$$\Pr\left(X_i\right)=\Pr\left(\sum_{j=1}^{N/B}d_{ij}\geq\frac{P_1}{2}\right)\leq\Pr\left(\sum_{j=1}^{N/B}d'_{ij}\geq\frac{P_1}{2}\right).$$

Since the expected value of the sum of $\{d'_{ij}\}$ is $\frac{s}{B-1}$, the Chernoff bound is suitable for this case, and we find

$$\begin{aligned}\Pr\left(X_i\right)\leq&\Pr\left(\sum_{j=1}^{N/B}d'_{ij}-\frac{s}{B-1}\geq\frac{P_1}{2}-\frac{s}{B-1}\right)\\ \leq&\mathrm{e}^{-\frac{[(B-1)P_1-2s]^2}{2(B-1)[(B-1)P_1+2s]}}.\end{aligned}$$

According to the union bound, the event $X$ which denotes that there exist some left nodes with degree larger than $\frac{P_1}{2}$ follows from the upper bound,

$$\Pr\left(X\right)\leq BC\Pr\left(X_i\right)\leq BC\mathrm{e}^{-\frac{[(B-1)P_1-2s]^2}{2(B-1)[(B-1)P_1+2s]}}.$$

That this is at most $\mathrm{e}^{-O(n)}$ follows since $B = \Theta(s)$, $s = \Theta(N^\delta)$, and $P_1 = \Theta(n)$. □

The degree bound we have just proven allows us to bound the maximum element of the array $\mathbf{T}$.

**Lemma 9.** *Suppose the maximum degree of the right nodes in $G$ is not larger than $\frac{P_1}{2}$. Then the maximum element of the array $\mathbf{T}$ is at most 4 if all the earlier bin detectors work correctly.*

*Proof.* The recursive equation for $\mathbf{T}$ in Algorithm 3 is

$$\mathbf{T}_{c'}[j_{c'}] \leftarrow \mathbf{T}_{c'}[j_{c'}] + \frac{\mathbf{T}_c[j]}{P_1} + \frac{(P_1-1)B}{P_1 N}.$$

There exists a time sequential order for each nonzero Pauli error rate to be detected. For each step $i$ and bin $\mathbf{U}_{z_i}$, being the next bin in which we will find a nonzero Pauli rate. Let $T_{max}$ denote the current maximum element in a subset $\mathbf{T}_{peeled}$ of $\mathbf{T}$. $\mathbf{T}_{peeled}$ contains $T$ of bins in which we have already found a nonzero Pauli error. Also, we use $T'_{max}$ to indicate the maximum T that will exist after the next step.

According to the assumption, the maximum degree of an arbitrary right node is less than $\min(P_1/2, N/B)$, and the number of peels needed is never more than the maximum degree of that node. Note our assumption is that the previous bins have been accurately detected, so the process will always choose nonzero Pauli error rates (and the corresponding bins) to peel. The noise in the peeling bin will increase by at most $\frac{T_{\max}}{P_1} + \frac{(P_1-1)B}{P_1 N}$. Each $\mathbf{T}$ is initialized as 1 and denote the number of peeling by $\kappa$, therefore

$$\begin{aligned} T'_{\max} &\le 1 + \kappa \cdot \left( \frac{T_{\max}}{P_1} + \frac{(P_1-1)B}{P_1 N} \right) \\ &\le 1 + \frac{T_{\max}}{2} + 1 \end{aligned}$$

Then by induction, because initially the maximum element in $\mathbf{T}_{peeled}$ is equal to 0 and in each step this value will increase as the above formula. Therefore, the maximum element we can get is the limit of this equation, which is $T_{max} \le 4$. □

According to the above two lemmas, the integer $T_{\max}$ indicates that the upper bound of the maximum element in the array $\mathbf{T}$ is not larger than 4 with high probability for a sufficiently large $N$ . This fact will help us to prove an exponentially decaying bound on the failure probability.

In order to compute the failure rates and make the algorithm realizable, we assumed (as in **A3**) the minimum nonzero Pauli error rate $\rho_0$ satisfies $\rho_0^2 \ge 4\nu^2 = 4\xi^2/B$. In anticipation of applying the union bound, let us define the following error categories and their probabilities. For brevity, we will denote a zero-ton, single-ton, or multi-ton by just the letters $\mathtt{z}$, $\mathtt{s}$, or $\mathtt{m}$, and we denote the true value of the bin by $\mathfrak{B}$.

**Definition 5** (Failure modes for bin detection)**.** *The bin detection algorithm failure modes are defined as follows:*

- *The single-ton false negative probability:*

$$\Pr(\mathrm{SFN}) := \Pr(\widehat{\mathfrak{B}} = \mathtt{z} \,|\, \mathfrak{B} = \mathtt{s}) + \Pr(\widehat{\mathfrak{B}} = \mathtt{m} \,|\, \mathfrak{B} = \mathtt{s})$$

- *The single-ton false positive probability:*

$$\Pr(\mathrm{SFP}) := \Pr(\widehat{\mathfrak{B}} = \mathtt{s} \,|\, \mathfrak{B} = \mathtt{z}) + \Pr(\widehat{\mathfrak{B}} = \mathtt{s} \,|\, \mathfrak{B} = \mathtt{m})$$

- *The multi-ton $\leftrightarrow$ zero-ton confusion probability:*

$$\Pr(\mathrm{MZ}) := \Pr(\widehat{\mathfrak{B}} = \mathtt{z} \,|\, \mathfrak{B} = \mathtt{m}) + \Pr(\widehat{\mathfrak{B}} = \mathtt{m} \,|\, \mathfrak{B} = \mathtt{z})$$

- *The index error probability:*

$$\Pr(\mathrm{I}) := \Pr(\widehat{\mathfrak{B}} = \mathtt{s}, \widehat{m} \neq m \,|\, \mathfrak{B} = \mathtt{s}, m)$$

- *The value error probability:*

$$\Pr(\mathrm{V}) := \Pr(\widehat{\mathfrak{B}} = \mathtt{s}, |\hat{p}_{\hat{m}} - p_m| > \sqrt{4\nu^2} \,|\, \mathfrak{B} = \mathtt{s}, m, p_m)$$

Of course these probabilities are not all independent. However, by the union bound it suffices to bound each of these bad events individually and the total failure probability will be at most the sum of the probabilities of these failure modes. We will show that all of these failure probabilities decay exponentially with $P_1$, the number of randomly chosen offsets.

**Lemma 10.** *Let $E$ denote the event that an arbitrary bin detection with inputs as those in Algorithm 3 returns either the wrong bin type, the wrong index or an estimated Pauli error rate with error larger than $\sqrt{4\nu^2}$. Let $D$ be the event that the maximum degree of right nodes in $G$ is not larger than $P_1/2$. Let $V$ be the event that all the earlier bin detectors work correctly. And denote the event that every peeling route is cycle-free by $H$. There exists a bound that*

$$\Pr(E|D,V,H) \le \mathrm{e}^{-O(n)}. \tag{F10}$$

*Proof.* This theorem means that the bin detector algorithm succeeds with high probability whenever $D$, $V$ and $H$ occur. To show it, we have to bound the failure probabilities for each failure mode of the bin detection algorithm and then implement the union bound. We will prove most of our statements by bounding failure probabilities with expressions of the form $\mathrm{e}^{-O(P_1)}$. This is equivalent to a bound of the form $\mathrm{e}^{-O(n)}$ since $P_1 = \Theta(n)$ by Definition 3. Also note that conditioning on events $D$, $V$ and $H$ allows the use of Lemma 7 and Lemma 9, specifically that the variance of noise in each row of this bin is $T\nu^2$ from Lemma 7, and that this $T$ is no more than 4 according to Lemma 9.

We first consider the *single-ton false negative* probability in Definition 5. Note in this case, the underlying

bin contains only one Pauli error rate along with noise, that is

$$\mathbf{U}_c = p_m(-1)^{\langle \mathbf{D}_c, m\rangle} + \mathbf{W}. \tag{F11}$$

Let $f_1 = \Pr\left(\widehat{\mathfrak{B}} = \mathbf{z} \,|\, \mathfrak{B} = \mathbf{s}\right)$. Then by definition, the probability can be upper bounded by the probability of a single-ton bin passing the zero-ton verification:

$$f_1 \leq \Pr\left(\frac{1}{P_1}\left\|p_m \mathbf{s}_{c,m} + \mathbf{W}\right\|^2 \leq T(1+\gamma_1)\nu^2\right),$$

where $\mathbf{s}_{c,m}$ is the vector with $t$-th element to be the random sign of index $m$ and the offset $d_{c;t}$. Since here the noise vector $\mathbf{W}$ comes from the sum of noise $w$, it's obvious that all the elements of $\mathbf{W}$ are Gaussian distributed with variance $T\nu^2$.

Therefore, according to the tail bounds of Lemma 4, the following holds as long as $\gamma_1 < \rho_0^2/T\nu^2$.

$$f_1 \leq \mathrm{e}^{-\frac{P_1}{4}\frac{\left(\rho_0^2/T\nu^2-\gamma_1\right)^2}{1+2\rho_0^2/T\nu^2}}. \tag{F12}$$

Now let $f_2 = \Pr\left(\widehat{\mathfrak{B}} = \mathtt{m} | \mathfrak{B} = \mathbf{s}\right)$. This kind of failure happens if and only if the single-ton bin fails during single-ton verification,

$$f_2 = \Pr\left(\frac{1}{P_1}\left\|\mathbf{U}_c - \widehat{p}_{\widehat{m}}\mathbf{s}_{c,\widehat{m}}\right\|^2 \geq T(1+\gamma_2)\nu^2\right).$$

Considering the underlying structure of this bin (F11), this probability can be bounded using a conditional probability. We first denote the event $\{|\widehat{p}_{\widehat{m}} - p_m| > \sqrt{4\nu^2}$ or $\widehat{m} \neq m\}$ by $E_0$. Then we observe

$$f_2 \leq \Pr(E_0) + \Pr\left(\frac{1}{P_1}\left\|\mathbf{U}_c - \widehat{p}_{\widehat{m}}\mathbf{s}_{c,\widehat{m}}\right\|^2 \geq T(1+\gamma_2)\nu^2 \middle| E_0^c\right).$$

Using the tail bound (F1), we have that

$$\begin{aligned}&\Pr\left(\frac{1}{P_1}\left\|\mathbf{U}_c - \widehat{p}_{\widehat{m}}\mathbf{s}_{c,\widehat{m}}\right\|^2 \geq T(1+\gamma_2)\nu^2 \middle| E_0^c\right)\\ &\leq \mathrm{e}^{-\frac{P_1}{4}\left(\sqrt{1+2\gamma_2}-\sqrt{1+2\times 4/T}\right)^2}.\end{aligned} \tag{F13}$$

Then using union bound, we can deal with the first term

$$\begin{aligned}\Pr(E_0) &\leq \Pr\left(|\widehat{p}_{\widehat{m}} - p_m| > \sqrt{4\nu^2}\right) + \Pr(\widehat{m} \neq m)\\ &\leq \Pr\left(|\widehat{p}_{\widehat{m}} - p_m| > \sqrt{4\nu^2} \middle| \widehat{m} = m\right) + 2\Pr(\widehat{m} \neq m).\end{aligned} \tag{F14}$$

Note above, the estimated Pauli error rate can be calculated according to Algorithm 4, so we obtain the bound

$$\begin{aligned}&\Pr\left(|\widehat{p}_{\widehat{m}} - p_m| > \sqrt{4\nu^2} \middle| \widehat{m} = m\right)\\ =&\Pr\left(\left|\frac{\mathbf{s}_{c,m}^T \mathbf{U}_c}{P_1} - p_m\right| > \sqrt{4\nu^2}\right)\\ =&\Pr\left(|Y|/P_1 > \sqrt{4\nu^2}\right) \leq 2\mathrm{e}^{-\frac{4P_1}{2T}},\end{aligned} \tag{F15}$$

where $Y$ is the sum of $P_1$ i.i.d. Gaussian variables with $\mathcal{N}\left(0, \left(T + \frac{(P_1-1)B}{N}\right)\nu^2\right)$ like (F7), and the last inequality comes from the Chernoff-Hoeffding bound [32]. According to Lemma 9 powers in (F13) and (F15) are both scaling linearly with $P_1$, thus the probabilities decay exponentially with $P_1$.

Since the second term in (F14), $\Pr(\widehat{m} \neq m)$, is essentially the probability of the index error, the failure probability of such a decoding process also decays exponentially with $P_2$. In accordance with (D10), the sign vector $[\mathsf{sgn}[U_{P_1}], \cdots, \mathsf{sgn}[U_{P-1}]]^T$ is the sum of a codeword $\langle \mathbf{G}, m\rangle$ and a vector of noise. Since the decoding process fails only if the weight of the noise is larger than distance $\beta P_2$ and each element of the noise is an independent Bernoulli random variable with error probability upper bounded by $\mathbb{P}$, the index error probability can be bounded by Chernoff-Hoeffding bound:

$$\Pr(\widehat{m} = m) \leq \mathrm{e}^{-\frac{(\beta/\mathbb{P}-1)^2}{3}P_2}. \tag{F16}$$

Moreover, as noted in Remark 1, we have $\mathbb{P}_m < \frac{1}{2}$ for all $m \in \mathbb{F}_2^{2n}$. Given the assumptions of $D$, $V$ and $H$, we choose the maximum one to be $\mathbb{P}$. Therefore, using the law of total probability we have

$$\begin{aligned}f_2 \leq &\mathrm{e}^{-\frac{P_1}{4}\left(\sqrt{1+2\gamma_2}-\sqrt{1+2\times 4/T}\right)^2} + 2\mathrm{e}^{-\frac{4P_1}{2T}}\\ &+ 2\mathrm{e}^{-\frac{(\beta/\mathbb{P}-1)^2}{3}P_2}.\end{aligned} \tag{F17}$$

Recall the Definition 3 to learn the magnitude of $P_1, P_2$, and we have a bound $f_2 = \mathrm{e}^{-O(n)}$.

We now turn to the case that the bin detection algorithm incorrectly recognizes a zero-ton or a multi-ton bin as a single-ton bin, i.e., we consider the *single-ton false positive* probability. For this, we need to consider the general underlying bin structure

$$\mathbf{U}_c = \mathbf{S}_c\mathbf{p} + \mathbf{W}, \tag{F18}$$

where $\mathbf{U}_c$ is either zero-ton or multi-ton, and only contains the $P_1$ fully random offsets when choosing as Definition 3. Here $\mathbf{S}_c \in \{\pm 1\}^{P_1 \times N/B}$ is the sign matrix constructed according to Lemma 2.

Now consider the probability of the bin detector falsely detecting a zero-ton as a single-ton, and denote $\Pr\left(\widehat{\mathfrak{B}} = \mathbf{s} | \mathfrak{B} = \mathbf{z}\right)$ by $f_3$. By definition, the probability of $f_3$ can be bounded by the probability of zero-ton verification failing

$$f_3 \leq \Pr\left(\frac{1}{P_1}\left\|\mathbf{W}\right\|^2 \geq T \times (1+\gamma_1)\nu^2\right).$$

According to the tail bound (F1), this failure probability can be bounded by an exponentially decaying function

$$f_3 \leq \mathrm{e}^{-\frac{P_1}{4}(\sqrt{1+2\gamma_1}-1)^2}. \tag{F19}$$

Now let $f_4 = \Pr\left(\widehat{\mathfrak{B}} = \mathfrak{s} | \mathfrak{B} = \mathfrak{m}\right)$. The kind of error probability can be evaluated under the multi-ton model when it passes the single-ton verification step for some estimated index-value pair $(\widehat{m}, \widehat{p}_{\widehat{m}})$

$$f_4 \leq \Pr\left(\frac{1}{P_1}\left\|\mathbf{U}_c - \widehat{p}_{\widehat{m}}\mathbf{s}_{c,\widehat{m}}\right\|^2 \leq T \times (1+\gamma_2)\nu^2\right).$$

Let

$$\mathbf{g}_c = \mathbf{S}_c\mathbf{p} - \widehat{p}_{\widehat{m}}\mathbf{s}_{c,\widehat{m}}, \tag{F20}$$

and let the sample error be $\mathbf{W} = \mathbf{k}$. Then the law of total probability can be used as follows:

$$\begin{aligned} f_4 =& \Pr\left(\frac{1}{P_1}\left\|\mathbf{g}+\mathbf{k}\right\|^2 \leq T(1+\gamma_2)\nu^2\right) \\ \leq& \Pr\left(\frac{1}{P_1}\left\|\mathbf{g}+\mathbf{k}\right\|^2 \leq T(1+\gamma_2)\nu^2 \middle| \frac{\|\mathbf{g}\|^2}{P_1} \geq 2T\gamma_2\nu^2\right) \\ &+ \Pr\left(\frac{\|\mathbf{g}\|^2}{P_1} \leq 2T\gamma_2\nu^2\right). \end{aligned} \tag{F21}$$

Note that the first term can be bounded by (F2) since the conditional part shows the lower bound of the parameter $\theta_0$ as defined in Lemma 4

$$\begin{aligned} &\Pr\left(\frac{1}{P_1}\left\|\mathbf{g}+\mathbf{k}\right\|^2 \leq T(1+\gamma_2)\nu^2 \middle| \frac{\|\mathbf{g}\|^2}{P_1} \geq 2T\gamma_2\nu^2\right) \\ &\leq \mathrm{e}^{-\frac{P_1}{4}\frac{\gamma_2^2}{1+4\gamma_2}}. \end{aligned} \tag{F22}$$

The second term can be bounded as follows. Let $\alpha = \mathbf{p} - \widehat{p}_{\widehat{m}}\mathbf{e}_{\widehat{m}}$, and we have

$$\Pr\left(\frac{\|\mathbf{g}\|^2}{P_1} \leq 2T\gamma_2\nu^2\right) = \Pr\left(\frac{\|\mathbf{S}_c\alpha\|^2}{P_1} \leq 2T\gamma_2\nu^2\right).$$

We denote the support set of the vector $\alpha$ by $\mathcal{L}_0$, and define the $\rho_0$-essential support of $\alpha$ to be

$$\mathcal{L} = \left\{i \in \mathbb{F}_2^{2n} \,\middle|\, |\alpha_i| \geq \rho_0\right\}. \tag{F23}$$

Denote the cardinality of $\mathcal{L}$ by $L$. Then the above probability can be bounded by an application of the Chernoff-Hoeffding bound.

With the same argument as in the proof of Lemma 6, the sample error in each row in vector $\mathbf{g}$ is independent, and so is the square of that error. When we calculate $\|\mathbf{g}\|^2$, we can regard it as a sum of $P_1$ independent random variables. Also, each term in this sum contains the same structure, and identically distributed parameters, so we can claim each term is identically distributed.

Therefore, we first analyze the expected value $E$ of each variable in this sum. Take one of these terms $X_i$ as an example,

$$X_i := \left(\sum_{j\in\mathcal{L}_0}(-1)^{\langle d_i,j\rangle}\alpha_j\right)^2, \tag{F24}$$

where $\{d_i\}$ is a set of independent random $2n$-bit strings. The expected value $E$ of $X_i$ satisfies the following bound

$$E = \mathbb{E}(X_i) \geq L\rho_0^2. \tag{F25}$$

Note that above we used the fact that any random strings are independent.

Moreover, since we want to show this term will be large with high probability, we should consider the random terms in each $X_i$, and that is

$$R_i = \sum_{\substack{u>v \\ u,v\in\mathcal{L}_0}}(-1)^{\langle d_i,u+v\rangle}2\alpha_u\alpha_v, \tag{F26}$$

where the order is in lexicographical order. Note the rest part of $X_i$ is a deterministic one, so we only calculate the variance for this $R_i$. It's straightforward that we have

$$\mathrm{Var}\left(R_i\right) = \mathbb{E}[R_i^2], \tag{F27}$$

and the only contributed terms are those without random signs in $R_i^2$. For example, if we consider a specific $u, v$ and the term $(\alpha_u\alpha_v) \times (\alpha_w\alpha_x)$ with some $(w, x) \neq (u, v)$ (assume $w > x$), the probability that the sign is always $+1$ is no larger than $\frac{L^2}{4^n}$ (here we handle the probability as the different Paulis occur evenly). Moreover, consider there are altogether $\frac{L(L-1)}{2}$ terms in $R_i$ so the probability is as follows

$$\Pr\left(\mathrm{Var}\left(R_i\right) = \sum_{\substack{u>v \\ u,v\in\mathcal{L}_0}}4\alpha_u^2\alpha_v^2\right) \geq 1 - O\left(\frac{L^4}{4^n}\right). \tag{F28}$$

Then we can use the Hoeffding bound to obtain

$$\begin{aligned} &\Pr\left(\frac{\|\mathbf{g}\|^2}{P_1} \leq 2T\gamma_2\nu^2\right) = \Pr\left(\frac{\sum_{i=0}^{P_1-1}X_i}{P_1} \leq 2T\gamma_2\nu^2\right) \\ &= \Pr\left(E - \frac{\sum_{i=0}^{P_1-1}X_i}{P_1} \geq E - 2T\gamma_2\nu^2\right) \\ &\leq \exp\left(-\frac{P_1(L\rho_0^2 - 2T\gamma_2\nu^2)^2}{2L^2\rho_0^4}\right) + O\left(\frac{P_1L^4}{4^n}\right). \end{aligned} \tag{F29}$$

The last inequality used the total probability and that $\mathbb{E}[X_i]^2 \geq \sum_{\substack{u>v \\ u,v\in\mathcal{L}_0}}4\alpha_u^2\alpha_v^2$.

For any nontrivial signal, we have that $1 < L < P_1/2$. As long as $0 < \gamma_2 < \rho_0^2/2T\nu^2$, for any multi-ton there

exists

$$f_4 \leq \mathrm{e}^{-\frac{P_1}{4}\frac{\gamma_2^2}{1+4\gamma_2}} + \mathrm{e}^{-\frac{P_1(\rho_0^2-2T\gamma_2\nu^2)^2}{2\rho_0^4}} + O\left(\frac{P_1^5}{N}\right). \quad \text{(F30)}$$

Next we will consider the *multi-ton–zero-ton confusion* probability

$$\Pr(\mathrm{MZ}) := \Pr(\widehat{\mathfrak{B}} = \mathtt{z} \,|\, \mathfrak{B} = \mathtt{m}) + \Pr(\widehat{\mathfrak{B}} = \mathtt{m} \,|\, \mathfrak{B} = \mathtt{z}).$$

Denote the first term $\Pr(\widehat{\mathfrak{B}} = \mathtt{z} \,|\, \mathfrak{B} = \mathtt{m})$ by $f_5$ and the second $\Pr(\widehat{\mathfrak{B}} = \mathtt{m} \,|\, \mathfrak{B} = \mathtt{z})$ by $f_6$. For $f_5$, recognizing a multi-ton as a zero-ton, we have the following inequality,

$$f_5 \leq \Pr\left(\frac{1}{P_1}\|\mathbf{U}\|^2 \leq T \times (1+\gamma_1)\nu^2\right).$$

Note this probability can be analysed in just the same way as $f_4$'s, and the only difference is that when we consider $f_5$, the $\alpha$ just based on several underlying Pauli error rates without any subtraction, so $L \geq 2$ for this case. As long as $0 < \gamma_1 < \rho_0^2/T\nu^2$, for any multi-ton we have the bound

$$f_5 \leq \mathrm{e}^{-\frac{P_1}{4}\frac{\gamma_1^2}{1+4\gamma_1}} + \mathrm{e}^{-P_1\frac{(2\rho_0^2-2T\gamma_1\nu^2)^2}{8\rho_0^4}} + O\left(\frac{P_1^5}{N}\right). \quad \text{(F31)}$$

Moreover, it is clear that the failure probability of recognizing a zero-ton bin as a multi-ton bin, namely $f_6$, is smaller than $f_3$.

Next, consider the *index error* probability, and denote $\Pr\left(\widehat{\mathfrak{B}} = \mathtt{s}, \widehat{m} \neq m | \mathfrak{B} = \mathtt{s}, m\right)$ by $f_7$. This probability can be bounded by the probability of estimating a wrong index $\widehat{m}$ and some Pauli error rate, and still passing the single-ton verification

$$\begin{aligned} f_7 &\leq \Pr\left((\widehat{m} \neq m) \ \wedge \ (\widehat{m}, \widehat{p}_{\widehat{m}}) \text{ passes verification}\right) \\ &\leq \Pr\left(\widehat{m} \neq m\right) \leq \mathrm{e}^{-\frac{(\beta/\mathbb{P}-1)^2}{3}P_2}. \end{aligned} \quad \text{(F32)}$$

Note the last inequality is just (F16), and according to Remark 1, $\mathbb{P}_m < \frac{1}{2}$ for all $m \in \mathbb{F}_2^{2n}$ given that events $D$ and $V$ happen and we choose the maximum one to be $\mathbb{P}$.

Finally, let's consider the *value error* probability, and denote $\Pr(\widehat{\mathfrak{B}} = \mathtt{s}, |\hat{p}_{\widehat{m}} - p_m| > \sqrt{4\nu^2} | \mathfrak{B} = \mathtt{s}, m, p_m)$ by $f_8$. Note that we have chosen $\sqrt{4\nu^2}$ as the error bound for the Pauli error rate, so similar to the *index error* probability, this $f_8$ can be bounded by the probability of estimating a noisy Pauli error rate and passing the single-ton verification. We can loosen this bound by only consider the first event, and we obtain the inequality

$$\begin{aligned} f_8 &\leq \Pr\left(|\hat{p}_{\widehat{m}} - p_m| > \sqrt{4\nu^2} \middle| \widehat{m} = m\right) + \Pr\left(\widehat{m} \neq m\right) \\ &\leq 2\mathrm{e}^{-\frac{4P_1}{2T}} + \mathrm{e}^{-\frac{(\beta/\mathbb{P}-1)^2}{3}P_2}. \end{aligned} \quad \text{(F33)}$$

Note the last inequality comes from combination of (F15) and (F16). According to Remark 1, we again have $\mathbb{P}_m < \frac{1}{2}$ for all $m \in \mathbb{F}_2^{2n}$ given that events $D$, $V$ and $H$ happen and we choose the maximum one to be $\mathbb{P}$.

According to Definition 5, we have treated all of the failure cases of the bin detector algorithm. Using the union bound, we can get the following inequality

$$\Pr\left(E\right) \leq \sum_{i=1}^{8} f_i.$$

As we illustrated at the beginning of this proof, events $D$, $V$ and $H$ have showed that the variance of noise in each row of this bin is $T\nu^2$ from Lemma 7, and that this $T$ is no more than 4 according to Lemma 9. Furthermore, they imply that $\theta_m$ is strictly smaller than $\frac{1}{2}$ for all $m \in \mathbb{F}_2^{2n}$ in (F17), (F32) and (F33). Since constraining the peeling graph $G$ to obey this event is independent of the above analysis of the failure probabilities of the bin detector, it will hold that

$$\Pr\left(E | D, V, H\right) \leq \mathrm{e}^{-O(P_1)}.$$

This completes the proof. □


[1] J. M. Martinis, npj Quantum Information **1**, (2015), arXiv:1510.01406.

[2] J. Emerson, R. Alicki, and K. Życzkowski, J. Opt. B **7**, S347 (2005), quant-ph/0503243.

[3] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, Phys. Rev. A **77**, 012307 (2008), arXiv:0707.0963.

[4] S. T. Flammia and Y.-K. Liu, Phys. Rev. Lett. **106**, 230501 (2011), arXiv:1104.4695.

[5] J. Helsen, X. Xue, L. M. Vandersypen, and S. Wehner, npj Quantum Inf. **5**, 71 (2019), arXiv:1806.02048.

[6] T. J. Proctor, A. Carignan-Dugas, K. Rudinger, E. Nielsen, R. Blume-Kohout, and K. Young, Phys. Rev. Lett. **123**, 030503 (2019), arXiv:1807.07975.

[7] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, arXiv e-prints , arXiv:1902.08543 (2019), arXiv:1902.08543 [quant-ph].

[8] M. Sarovar, T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout, arXiv e-prints , arXiv:1908.09855 (2019), arXiv:1908.09855 [quant-ph].

[9] P. Aliferis and J. Preskill, Phys. Rev. A **78**, 052331 (2008), arXiv:0710.1301.

[10] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, Phys. Rev. Lett. **120**, 050505 (2018), arXiv:1708.08474.

[11] S. Puri, L. St-Jean, J. A. Gross, A. Grimm, N. E. Frattini, P. S. Iyer, A. Krishna, S. Touzard, L. Jiang, A. Blais,

S. T. Flammia, and S. M. Girvin, arXiv e-prints , arXiv:1905.00450 (2019), arXiv:1905.00450 [quant-ph].
[12] D. K. Tuckett, A. S. Darmawan, C. T. Chubb, S. Bravyi, S. D. Bartlett, and S. T. Flammia, Phys. Rev. X **9**, 041031 (2019), arXiv:1812.08186.
[13] D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, arXiv e-prints , arXiv:1907.02554 (2019), arXiv:1907.02554 [quant-ph].
[14] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Phys. Rev. Lett. **105**, 150401 (2010), arXiv:0909.3304.
[15] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White, Phys. Rev. Lett. **106**, 100401 (2011), arXiv:0910.5498.
[16] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert, New J. Phys. **14**, 095022 (2012), arXiv:1205.2300.
[17] A. V. Rodionov, A. Veitia, R. Barends, J. Kelly, D. Sank, J. Wenner, J. M. Martinis, R. L. Kosut, and A. N. Korotkov, Phys. Rev. B **90**, 144504 (2014), arXiv:1407.0761.
[18] A. Kalev, R. L. Kosut, and I. H. Deutsch, npj Quantum Information **1** (2015), 10.1038/npjqi.2015.18, arXiv:1502.00536.
[19] C. A. Riofrío, D. Gross, S. T. Flammia, T. Monz, D. Nigg, R. Blatt, and J. Eisert, Nature Communications **8**, 15305 (2017).
[20] I. L. Chuang and M. A. Nielsen, J. Mod. Opt. **44**, 2455 (1997), quant-ph/9610001.
[21] E. Knill, Nature **434**, 39 (2005), arXiv:quant-ph/0410199.
[22] J. J. Wallman and J. Emerson, Phys. Rev. A **94**, 052325 (2016), arXiv:1512.01098.
[23] M. Ware, G. Ribeill, D. Riste, C. A. Ryan, B. Johnson, and M. P. da Silva, arXiv preprint (2018), 1803.01818.
[24] S. T. Flammia and J. J. Wallman, "Efficient estimation of Pauli channels," (2019), 1907.12976.
[25] R. Harper, S. T. Flammia, and J. J. Wallman, arXiv preprint (2019), arXiv:1907.13022.
[26] E. Kushilevitz and Y. Mansour, SIAM Journal on Computing **22**, 1331 (1993).
[27] X. Li, J. K. Bradley, S. Pawar, and K. Ramchandran, in *2014 IEEE International Symposium on Information Theory* (IEEE, 2014) arXiv:1508.06336.
[28] R. Scheibler, S. Haghighatshoar, and M. Vetterli, IEEE Transactions on Information Theory **61**, 2115 (2015), arXiv:1310.1803.
[29] R. Koenig and J. A. Smolin, Journal of Mathematical Physics **55**, 122202 (2014), https://doi.org/10.1063/1.4903507.
[30] W. Feller, *An introduction to probability theory and its applications*, Vol. 2 (John Wiley & Sons, 2008).
[31] M. Sipser and D. A. Spielman, IEEE transactions on Information Theory **42**, 1710 (1996).
[32] W. Hoeffding, Journal of the American Statistical Association **58**, 13 (1963).