Improving error suppression with noise-aware decoding

arXiv:2502.21044

Evan T. Hockings Andrew C. Doherty Robin Harper

The University of Sydney

March 26, 2025

The backlog problem

- Fault tolerance is required for useful quantum computation.
- Real-time decoding is essential: syndrome data must be processed before implementing a non-Clifford operation.
- Seek techniques for improving decoder performance at scale without increasing computational cost.
- We introduce one such technique, *noise-aware decoding*, which uses noise estimates to calibrate decoders, and investigate it through numerical simulations.

A review of quantum error correction

Pauli operators

- The single-qubit Pauli operators are Hermitian, unitary, and hence involutions, span $\mathbb{C}^{2\times 2}$, and are given

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \; Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

- The n-qubit Pauli operators are n-fold tensor products.
- They form an orthogonal basis for $\mathbb{C}^{d\times d}$, where $d = 2^n$, under the natural *trace* or *Hilbert-Schmidt* inner product

$$\langle A, B\rangle = \operatorname{tr}\left(A^\dagger B\right).$$

- We can express n-qubit errors as a linear combination of n-qubit Pauli errors, enabling quantum error correction.

The Pauli group

- Index n-qubit Pauli operators by bit strings in $\mathbb{Z}_2^{2n}$, $\boldsymbol{a} = (\boldsymbol{a}^{(x)}, \boldsymbol{a}^{(z)}) = (a_1^{(x)}, \ldots, a_n^{(x)}, a_1^{(z)}, \ldots, a_n^{(z)})$, to write
$$P_{\boldsymbol{a}} = \bigotimes_{j=1}^{n} i^{a_j^{(x)} a_j^{(z)}} X^{a_j^{(x)}} Z^{a_j^{(z)}}.$$
- With phases $\langle i \rangle = \{\pm 1, \pm i\}$, these form the n-qubit *Pauli group* $\mathbf{P}^n$ under matrix multiplication.
- The Abelianisation $\mathrm{P}^n = \mathbf{P}^n / \langle i \rangle$, the *Pauli quotient group*, is isomorphic to $\mathbb{Z}_2^{2n}$ as, for $P_{\boldsymbol{a}}, P_{\boldsymbol{b}} \in \mathrm{P}^n$,
$$P_{\boldsymbol{a}} P_{\boldsymbol{b}} = P_{\boldsymbol{a}+\boldsymbol{b}}.$$
- For convenience, will refer to $\boldsymbol{a} \in \mathrm{P}^n$.

Pauli group commutation

- Define the commutation relation of Pauli operators with the *symplectic bilinear form* $\omega \colon \mathrm{P}^n \times \mathrm{P}^n \to \mathbb{Z}_2$,

$$\omega(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^{(x)} \cdot \boldsymbol{b}^{(z)} + \boldsymbol{a}^{(z)} \cdot \boldsymbol{b}^{(x)},$$

$$P_{\boldsymbol{a}} P_{\boldsymbol{b}} = (-1)^{\omega(\boldsymbol{a}, \boldsymbol{b})} P_{\boldsymbol{b}} P_{\boldsymbol{a}}.$$

- ω is *alternating*, $\omega(\boldsymbol{a}, \boldsymbol{a}) = 0$ for all $\boldsymbol{a}$, *non-degenerate*, $\omega(\boldsymbol{a}, \boldsymbol{b}) = 0$ for all $\boldsymbol{b}$ implies $\boldsymbol{a} = \boldsymbol{0}$, and symmetric as the field is $\mathbb{Z}_2$.
- Then (P^n, ω) is a *symplectic vector space.*
- It is convenient to play a little fast and loose with signs, though a more exacting treatment is possible.

The Clifford group

- The *Clifford group* is sometimes defined as the group of unitaries U that normalise the Pauli group $\mathbf{P}^n$, namely for any $P_{\boldsymbol{a}}$ there exists some $P_{\boldsymbol{b}}$ such that $UP_{\boldsymbol{a}}U^\dagger = P_{\boldsymbol{b}}$, but this has infinite centre with phases $e^{i\theta}$.
- Instead define the Clifford group $\mathbf{C}^n$ as the group generated by the Hadamard, phase, and controlled-X gates, written H_j, S_j, and $C_i(X_j)$, for control qubits i and target qubits $j \neq i$, where

$$H_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},\ S_1 = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix},\ C_1(X_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

- This yields 8 phases $\langle \eta \rangle$, where $\eta = \sqrt{i} = (1+i)/\sqrt{2}$.

Symplectic representation of the Clifford group

- The *Clifford quotient group* $\mathrm{C}^n = \mathbf{C}^n / \langle \eta, \mathbf{P}^n \rangle$ is isomorphic to the symplectic group $\mathrm{Sp}(2n, \mathbb{Z}_2)$, linear transformations on $\mathbb{Z}_2^{2n}$ that preserve ω, that is, for all $M \in \mathrm{C}^n$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathrm{P}^n$, $\omega(M\boldsymbol{a}, M\boldsymbol{b}) = \omega(\boldsymbol{a}, \boldsymbol{b})$.
- This *symplectic representation* of the Clifford group enables efficient simulation of *stabiliser circuits* with Clifford gates and computational basis measurements.
- Track states by their *stabiliser group* $S \subset \mathrm{P}^n$ such that $\omega(\boldsymbol{a}, \boldsymbol{b}) = 0$ for all $\boldsymbol{a}, \boldsymbol{b} \in S$.
- A state $|\psi\rangle$ is *stabilised* by S if $P_{\boldsymbol{a}}|\psi\rangle = |\psi\rangle$ for all $P_{\boldsymbol{a}} \in S$, and uniquely specified by S if it is *maximal*, or n-dimensional.

Pauli channels

- Model noise with a *Pauli channel*, which can be written

$$\mathcal{E}(\rho) = \sum_{\boldsymbol{a} \in \mathrm{P}^n} p_{\boldsymbol{a}} P_{\boldsymbol{a}} \rho P_{\boldsymbol{a}}.$$

- Learn $\mathcal{E}$ by estimating the 4^n *Pauli error probabilities* $p_{\boldsymbol{a}}$ that form a probability distribution over Pauli errors.
- The Pauli operators are the eigenvectors of $\mathcal{E}$

$$\mathcal{E}(P_{\boldsymbol{b}}) = \sum_{\boldsymbol{a} \in \mathrm{P}^n} p_{\boldsymbol{a}} P_{\boldsymbol{a}} P_{\boldsymbol{b}} P_{\boldsymbol{a}} = \left(\sum_{\boldsymbol{a} \in \mathrm{P}^n} (-1)^{\omega(\boldsymbol{a},\boldsymbol{b})} p_{\boldsymbol{a}} \right) P_{\boldsymbol{b}} = \lambda_{\boldsymbol{b}} P_{\boldsymbol{b}}.$$

- The *Pauli channel eigenvalues* $\lambda_{\boldsymbol{b}}$ are related to the error probabilities $p_{\boldsymbol{a}}$ by a *Walsh-Hadamard transform* ordered by ω, and more convenient to estimate.

Pauli channel estimation

- Consider the eigenbasis $|\psi_{\boldsymbol{s}}^{\boldsymbol{a}}\rangle$ of $P_{\boldsymbol{a}}$, sign configurations of tensor products of single-qubit Pauli eigenstates indexed by the length n bit string $\boldsymbol{s}$.
- Let s be the parity of $\boldsymbol{s}$, then $P_{\boldsymbol{a}}|\psi_{\boldsymbol{s}}^{\boldsymbol{a}}\rangle = (-1)^s|\psi_{\boldsymbol{s}}^{\boldsymbol{a}}\rangle$.
- Suppose we prepare eigenstates $|\psi_{\boldsymbol{s}}^{\boldsymbol{a}}\rangle$ of $P_{\boldsymbol{a}}$ uniformly at random, apply $\mathcal{E}$ m times, and measure the expectation value of $P_{\boldsymbol{a}}$, then

$$\frac{1}{2^n}\sum_{\boldsymbol{s}\in\mathbb{Z}_2^n}(-1)^s \operatorname{tr}\left(P_{\boldsymbol{a}}\mathcal{E}^m(|\psi_{\boldsymbol{s}}^{\boldsymbol{a}}\rangle\langle\psi_{\boldsymbol{s}}^{\boldsymbol{a}}|)\right) = \frac{1}{2^n}\operatorname{tr}\left(P_{\boldsymbol{a}}\mathcal{E}^m(P_{\boldsymbol{a}})\right) = \lambda_{\boldsymbol{a}}^m.$$

- This directly estimates $\lambda_{\boldsymbol{a}}^m$ and is the fundamental strategy underlying Pauli channel estimation techniques.

Pauli twirling

- Consider the *Pauli twirl* of a quantum channel $\mathcal{L}$,

$$\mathcal{L}^{\mathrm{P}^n}(\rho) = \frac{1}{4^n} \sum_{\boldsymbol{a} \in \mathrm{P}^n} \sum_k \left(P_{\boldsymbol{a}} L_k P_{\boldsymbol{a}}^\dagger\right) \rho \left(P_{\boldsymbol{a}} L_k P_{\boldsymbol{a}}^\dagger\right)^\dagger.$$

- Express L_k in terms of $P_{\boldsymbol{b}}$ with real coefficients $l_{k\boldsymbol{b}}$ as

$$L_k = \frac{1}{2^n} \sum_{\boldsymbol{b} \in \mathrm{P}^n} \operatorname{tr}\left(P_{\boldsymbol{b}}^\dagger L_k\right) P_{\boldsymbol{b}} = \sum_{\boldsymbol{b} \in \mathrm{P}^n} l_{k\boldsymbol{b}} P_{\boldsymbol{b}}.$$

- Calculate to find $\mathcal{L}^{\mathrm{P}^n}(\rho)$ is a Pauli channel with Pauli error probabilities

$$p_{\boldsymbol{b}} = \sum_k l_{k\boldsymbol{b}}^2.$$

- Hence *Pauli frame randomisation* and the *randomised compiling* protocol tailor quantum noise into Pauli noise.

Symplectic vector spaces

- Introduce stabiliser codes by first sketching results about symplectic vector spaces.
- Let V be a $2n$-dimensional vector space over the field F, and let $\omega\colon V \times V \to F$ be a symplectic bilinear form.
- The *symplectic complement* of a subspace $W \subseteq V$ is

$$W^\omega = \{v \in V : \forall w \in W, \omega(v, w) = 0\}.$$

- Then W is *isotropic* if $W \subseteq W^\omega$, *coisotropic* if $W^\omega \subseteq W$, and *Lagrangian* if $W = W^\omega$.
- The symplectic complement is the centraliser $C(S)$ of a subspace $S \subseteq \mathrm{P}^n$, stabiliser groups are isotropic, and maximal stabiliser groups are Lagrangian.

The rank-nullity theorem

- The *dual map* $\phi\colon V \to V^*$ acts as $\phi(v)w = \omega(w, v)$.
- For any subspace $W \subseteq V$, consider $\phi^{(W)}\colon V \to W^*$, where $\phi^{(W)}(v)w = \omega(w, v)$ for all $w \in W$.
- Since $\phi^{(W)}$ is surjective with kernel W^ω, the rank-nullity theorem yields

$$\dim W + \dim W^\omega = \dim V = 2n.$$

- This implies $W^{\omega\omega} = W$, so W is isotropic if and only if W^ω is coisotropic.
- Also isotropic subspaces have dimension at most n, and Lagrangian subspaces have dimension exactly n.

Symplectic bases

- Consider the basis $\{u_1, \ldots, u_n\}$ of a Lagrangian subspace L.
- This can be extended with $\{v_1, \ldots, v_n\}$ to obtain a *symplectic basis* for V with commutation properties

$$\omega(u_i, u_j) = \omega(v_i, v_j) = 0, \quad \omega(u_i, v_j) = \delta_{ij}, \quad \forall i, j \in [n].$$

- This follows from a symplectic Gram-Schmidt procedure, though the v_i are not unique.
- It is more efficient for stabiliser circuit simulations to track the entire symplectic basis.
- The u_i and v_i are called *stabiliser* and *destabiliser generators*, respectively.

Symplectic reductions

- Let $W \subseteq V$ be a coisotropic subspace and consider $\bar{W} = W/W^\omega$, the *symplectic reduction* of V by W.
- Then $\bar{\omega}([v],[w]) = \omega(v,w)$ is a well-defined symplectic form on $\bar{W}$, where $[w] = w + W^\omega \in \bar{W}$.
- Hence $(\bar{W}, \bar{\omega})$ is a symplectic vector space whose symplectic form $\bar{\omega}$ is inherited from ω on V.
- Also, let $L \subseteq W$ be a Lagrangian subspace of V, then $\bar{L} = L/W^\omega$ is a Lagrangian subspace of $\bar{W}$.
- Stabiliser codes are symplectic reductions of the Pauli group, which behave like smaller, redundantly encoded Pauli groups whose elements are the logical operators.

Stabiliser codes

- A *stabiliser code* encoding k logical qubits in n physical qubits is defined by a generating set $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ for a maximal stabiliser group, extended to a symplectic basis by $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}$.
- $S = \langle \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n-k} \rangle$ is generated by $n-k$ *stabiliser generators.*
- $L_S = \langle \boldsymbol{s}_{n-k+1}, \ldots, \boldsymbol{s}_n \rangle = \langle \bar{Z}_1, \ldots, \bar{Z}_k \rangle$ is generated by k *logical stabiliser generators.*
- $R = \langle \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{n-k} \rangle$ is generated by $n-k$ *destabiliser generators.*
- $L_R = \langle \boldsymbol{r}_{n-k+1}, \ldots, \boldsymbol{r}_n \rangle = \langle \bar{X}_1, \ldots, \bar{X}_k \rangle$ is generated by k *logical destabiliser generators.*

Stabiliser code distance

- Define the *logical group* $L = L_S \oplus L_R$ and partition the Pauli group as

$$\mathrm{P}^n = S \oplus L \oplus R.$$

- Then any $\boldsymbol{a} \in \mathrm{P}^n$ can be written as $\boldsymbol{a} = \boldsymbol{a}_S + \boldsymbol{a}_L + \boldsymbol{a}_R$ for $\boldsymbol{a}_S \in S$, $\boldsymbol{a}_L \in L$, and $\boldsymbol{a}_R \in R$.
- Also $C(S) = S \oplus L$, and logical operators are elements of the symplectic reduction $C(S)/S \cong L$.
- The *distance* of the code is the minimum weight non-trivial logical operator

$$d = \min_{\boldsymbol{a} \in C(S) \setminus S} |\boldsymbol{a}|.$$

Stabiliser codes under noise

- Suppose the n physical qubits are acted on by a Pauli channel $\mathcal{E}$ and some *physical error* $\boldsymbol{e} \in \mathrm{P}^n$ occurs, where $\boldsymbol{e} = \boldsymbol{e}_S + \boldsymbol{e}_L + \boldsymbol{e}_R$.
- Measure the stabiliser generators $\boldsymbol{s}_j$ for $j \in [n-k]$ with outcomes $(-1)^{s_j}$ for $s_j \in \mathbb{Z}_2$, giving the *error syndrome* $\boldsymbol{e}_R = s_1\boldsymbol{r}_1 + \cdots + s_{n-k}\boldsymbol{r}_{n-k} \in R$.
- Given $\mathcal{E}$ and $\boldsymbol{e}_R$, the problem of *decoding* the code is finding a *recovery operator* $\boldsymbol{f} \in \mathrm{P}^n$ such that $\boldsymbol{f} = \boldsymbol{e} + \boldsymbol{s}'$ for some $\boldsymbol{s}' \in S$.
- If the decoder succeeds, applying $\boldsymbol{f}$ corrects any logical errors, else the logical error specified by $\boldsymbol{e} + \boldsymbol{f}$ occurs.

Quantum error correction conditions

- The *quantum error correction conditions* on the *error set* $E \subseteq \mathrm{P}^n$ guarantee decoding success.
- For any error $\boldsymbol{e} \in E$, choose any recovery operator $\boldsymbol{f} \in E$ with appropriate error syndrome $\boldsymbol{f}_R = \boldsymbol{e}_R$, then $\boldsymbol{e} + \boldsymbol{f} = \boldsymbol{s}' + \boldsymbol{l}'$ for some $\boldsymbol{s}' \in S$ and $\boldsymbol{l}' \in L$.
- Decoding succeeds if $\boldsymbol{l}' = \mathbf{0}$, which is ensured by $\boldsymbol{e} + \boldsymbol{f} \notin C(S) \setminus S$.
- This implies decoding always succeeds if errors in E have weight at most $\lfloor (d-1)/2 \rfloor$.

Decoding strategies

- *Maximum-likelihood* decoding chooses the $\boldsymbol{f} \in \boldsymbol{l}' + \boldsymbol{r} + S$ with most probable $\boldsymbol{l}' \in L$ according to $\mathcal{E}$ given $\boldsymbol{r} \in R$, that is,

$$\boldsymbol{l}' = \arg\max_{\boldsymbol{m} \in L} \sum_{\boldsymbol{t} \in S} p_{\boldsymbol{t}+\boldsymbol{m}+\boldsymbol{r}}.$$

- *Minimum-weight* decoding chooses the most probable $\boldsymbol{s}' + \boldsymbol{l}' \in S \oplus L$ according to $\mathcal{E}$ given $\boldsymbol{r} \in R$, that is,

$$\boldsymbol{s}' + \boldsymbol{l}' = \arg\max_{\boldsymbol{t} \in S, \boldsymbol{m} \in L} p_{\boldsymbol{t}+\boldsymbol{m}+\boldsymbol{r}}.$$

- Decoder performance relies on knowledge of $\mathcal{E}$.
- We show that calibrating this *decoder prior* improves decoding performance.

The circuit-level picture of quantum error correction and fault tolerance

The ‘circuit-forward’ approach

- Google has demonstrated the surface code with many different syndrome extraction circuits.[1]
- I claim this reflects an emerging ‘circuit-forward’ paradigm focusing on the actual circuits run on the quantum device.
- This contrasts with a ‘code-forward’ paradigm that regards the design of quantum error correction circuits more as an implementation detail.
- Under the ‘circuit-forward’ paradigm, it becomes natural to co-design quantum error correcting codes, decoders, fault-tolerant circuits, and quantum devices.

[1] Google Quantum AI. Demonstrating dynamic surface codes. arXiv:2412.14360.

Open-source tools

- The 'circuit-forward' paradigm is powered by open-source packages such as Stim and PyMatching by Craig Gidney and Oscar Higgott—perhaps not coincidentally at Google.
- These enable stabiliser circuit simulation and decoding of quantum error correction circuits, respectively.
- But both simulation and decoding must be informed by a circuit-level Pauli noise model!
- My open-source package QuantumACES.jl enables the estimation of circuit-level Pauli noise at scale, which can inform simulation and decoding.
- This talk focuses on the latter.

The detector formalism

- Stim frames quantum error correction in terms of *detectors*, parities of measurement outcomes in a quantum error correction circuit that are deterministic absent noise.
- Also, *logical observables* are parities of measurement outcomes that correspond to logical Pauli operators.
- Errors flip detectors and logical observables.
- Given a circuit-level Pauli noise model, Stim constructs a *detector error model* describing the error probabilities of all possible combinations of detectors and logical observables.
- PyMatching uses the detector error model to decode the logical observables given the outcomes of the detectors.

Memory experiments

- Consider a Z (X) memory experiment.
- In the first round of syndrome extraction, the detectors are the Z-type (X-type) stabiliser measure qubit outcomes.
- In subsequent rounds, the detectors are both the Z- and X-type stabiliser measure qubit outcomes.
- In the final round, the detectors are parities of the Z-type (X-type) stabiliser measure qubit outcomes alongside the associated data qubit outcomes.
- The logical observable is the parity of data qubits in any logical Z (X) operator.

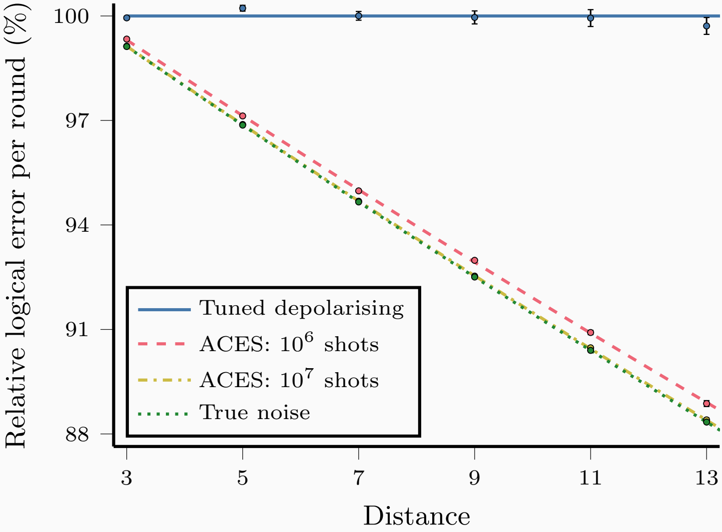
Noise-aware decoding

Calibrating decoders

- We use averaged circuit eigenvalue sampling (ACES) to characterise circuit-level Pauli noise in surface code syndrome extraction circuits,[2] implemented with QuantumACES.jl.
- Calibrating the PyMatching detector error model with ACES noise estimates enables noise-aware decoding.
- Below threshold, the logical error per round is approximately $\varepsilon \propto \Lambda^{-d/2}$.
- Noise-aware decoding increases the error suppression factor Λ, exponentially reducing logical error rates.

[2] Hockings, Doherty, Harper. Scalable noise characterization of syndrome-extraction circuits with averaged circuit eigenvalue sampling. PRX Quantum 6, 010334, 2025.

Noise-aware decoding



Noise-aware decoding at scale

- Trends are consistent with memory results at distance 25.

Decoder performance for memory experiments with 25 rounds, dividing 10^7 shots evenly between Z and X memory types. Diagonal elements count decoding failures for each prior. Off-diagonal elements count the number of shots where the decoder for the row succeeded and the decoder for the column failed.

Fail. Fail. / Succ.	True	ACES:10^7	ACES:10^6	Depolarising
True	**5507**	227	619	3005
ACES:10^7	195	**5539**	564	2997
ACES:10^6	495	472	**5631**	2994
Depolarising	1314	1338	1427	**7198**

Conclusions

- Noise-aware decoding can substantially reduce logical error rates and qubit overheads, with improvements that increase exponentially with scale.
- ACES noise estimates enable near-optimal decoding compared to calibration with the true noise model.
- In superconducting quantum computers, decoders could be calibrated with ACES experiments performed and processed in seconds!
- Now working to implement these methods on real quantum devices.