Implementing Distance-3 Quantum Codes with Noise in Qiskit

Bala (Dhruv Balasubramanian), Rodrigo (Walter Lopez)

December 17, 2025

Abstract

We implement a full quantum error correction pipeline for a family of $[[n, 1, 3]]$ codes with $n \in \{3, 5, 7\}$ using Qiskit. In each iteration of running the pipeline, we pick one of five input states, encode, apply one of four noise channels, measure stabilizers, apply conditioned correction gates, decode, and compute fidelity against the original state. We test four noise models (bit flip, phase flip, depolarizing, amplitude damping) with four error probabilities $p \in \{0.2, 0.4, 0.6, 0.8\}$. We compare each code to an unencoded baseline that experiences the same noise channel. Across most settings, especially for the 5 qubit and Steane codes, the mean fidelity advantage over baseline is negative. The results also show strong dependence on initial input state; some input states are almost invariant under the baseline channel, while others degrade. We use heatmaps at fixed p to display how each initial state is affected by each code.

1 Introduction

Quantum error correction is a way to solve for the problem that quantum information is fragile under noise. A quantum error correction code (QECC) defines a certain way to encode/decode quantum state as well as detect/correct errors. An error correction pipeline consists of the following steps: encode, noise, syndrome measurement, correction, decode. We run this pipeline on four $[[n, 1, 3]]$ codes: the 3 qubit bit-flip code, the 3 qubit phase-flip code, the perfect 5-qubit code, and the Steane 7-qubit code. We evaluate performance by measuring the fidelity between the decoded state and initial state. We run the same pipeline under four noise channels and four values of an error probability parameter p. We also compare to a baseline with no QECC encoding or recovery, where noise is applied to a single qubit without protection by a QECC.

2 Background

2.1 Definition of QECCs

A quantum error correcting code (QECC) with parameters $[[n, k, d]]$ encodes k logical qubits into n physical qubits and has distance d. Distance d implies detection of any error supported on fewer than d qubits, and correction of any error of weight at most $t = \lfloor (d-1)/2 \rfloor$ (given a valid error set).

In the stabilizer formalism, a code is specified by an Abelian subgroup $\mathcal{S}$ of the n qubit Pauli group that does not contain $-I$. The codespace is the simultaneous $+1$ eigenspace of all elements of $\mathcal{S}$. If $\{S_1, \ldots, S_{n-k}\}$ is an independent generating set, then measuring these generators gives us an $(n-k)$ bit syndrome.

For Pauli errors E, each syndrome bit records whether E commutes or anticommutes with a stabilizer generator:

$$s_i(E) = \begin{cases} 0, & [E, S_i] = 0, \\ 1, & \{E, S_i\} = 0. \end{cases} \tag{1}$$

An error set is correctable when the measured syndrome identifies a unique correction operation (up to stabilizer equivalence). In other words, within the error set we care about, each syndrome should map to one error gate that we apply to undo the error. If two distinct errors produce the same syndrome, then the code can detect that an error happened but cannot correct it.

Syndrome extraction uses ancillas and basis changes. To measure a Pauli product stabilizer, we rotate physical qubits so that each non-identity is transformed to the eigenbasis of Z and the eigenvalue parity is written onto an ancilla using CNOTs, then measure the ancilla.

2.2 Codes used

We compare four $[[n,1,3]]$ codes against an unencoded baseline.

3 qubit bit flip code $[[3,1,3]]$**.**

$$|0_L\rangle = |000\rangle, \qquad |1_L\rangle = |111\rangle. \tag{2}$$

A generating set is

$$S_1 = Z_1Z_2, \qquad S_2 = Z_2Z_3. \tag{3}$$

3 qubit phase flip code $[[3,1,3]]$**.**

$$|0_L\rangle = |+++\rangle, \qquad |1_L\rangle = |---\rangle. \tag{4}$$

A generating set is

$$S_1 = X_1X_2, \qquad S_2 = X_2X_3. \tag{5}$$

Perfect five qubit code $[[5,1,3]]$**.** A standard choice of generators is

$$S_1 = XZZXI, \quad S_2 = IXZZX, \quad S_3 = XIXZZ, \quad S_4 = ZXIXZ \tag{6}$$

with logical codewords

$$\begin{aligned}|0_L\rangle = \frac{1}{4}\Big(&|00000\rangle + |10010\rangle + |01001\rangle + |10100\rangle + |01010\rangle + |00101\rangle \\ &- |11011\rangle - |00110\rangle - |11000\rangle - |11101\rangle - |00011\rangle - |11110\rangle - |01111\rangle - |10001\rangle - |01100\rangle - |10111\rangle\Big),\end{aligned} \tag{7}$$

$$\begin{aligned}|1_L\rangle = \frac{1}{4}\Big(&|11111\rangle + |01101\rangle + |10110\rangle + |01011\rangle + |10101\rangle + |11010\rangle \\ &- |00100\rangle - |11001\rangle - |00111\rangle - |00010\rangle - |11100\rangle - |00001\rangle - |10000\rangle - |01110\rangle - |10011\rangle - |01000\rangle\Big).\end{aligned} \tag{8}$$

Steane code $[[7,1,3]]$**.** A generating set is

$$S_1^X = IIIXXXX, \qquad S_2^X = XIXIXIX, \qquad S_3^X = IXXIIXX, \tag{9}$$
$$S_1^Z = IIIZZZZ, \qquad S_2^Z = ZIZIZIZ, \qquad S_3^Z = IZZIIZZ \tag{10}$$

with logical codewords

$$|0_L\rangle = \frac{1}{\sqrt{8}}\Big(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle\Big), \tag{11}$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}\Big(|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle + |0101010\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle\Big). \tag{12}$$

3 Methods

We used Qiskit with the Aer `AerSimulator(method="density_matrix")` backend to run a full QEC pipeline: state preparation, encoding, a single application of a noise channel, stabilizer syndrome measurement with ancillas, syndrome conditioned correction gates, decoding, and fidelity evaluation against the original input state.

We tested five input states:

$$|0\rangle\,, \quad |1\rangle\,, \quad |+\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad |\psi_{\text{rand}}\rangle = a\,|0\rangle + b\,|1\rangle\,, \tag{13}$$

where (a, b) satisfy $|a|^2 + |b|^2 = 1$.

We generated $|\psi_{\text{rand}}\rangle$ by sampling $\theta \sim U[0, \pi]$, $\phi \sim U[0, 2\pi]$ and setting

$$a = \cos(\theta/2), \qquad b = e^{i\phi}\sin(\theta/2). \tag{14}$$

For each configuration, we saved the final single qubit density matrix ρ_{out} of the decoded logical qubit and computed the pure state fidelity

$$F(\rho_{\text{out}}, |\psi\rangle) = \langle\psi|\,\rho_{\text{out}}\,|\psi\rangle\,. \tag{15}$$

We compared the four QECC pipelines (bit flip, phase flip, $[[5, 1, 3]]$, Steane) against a no QECC baseline. The baseline prepares the same input state, applies the same single qubit noise channel once, and measures fidelity without encoding, syndrome extraction, correction, or decoding.

We tested four single qubit noise models with error probability parameter $p \in \{0.2, 0.4, 0.6, 0.8\}$. In each experiment, we applied the channel once to every physical qubit after encoding (and to the single qubit in the baseline case). Syndrome extraction, conditional recovery, and decoding are performed without additional noise. More realistic noise models would apply errors throughout the entire QEC circuit. We wrote `test_noise_models.py` to check the noise channels, and we include its output in the Appendix.

Below we list the channels tested along with their Kraus operator sum representations.

Bit flip.

$$\mathcal{E}_X(\rho) = (1-p)\rho + pX\rho X, \qquad K_0 = \sqrt{1-p}\,I, \quad K_1 = \sqrt{p}\,X. \tag{16}$$

Phase flip.

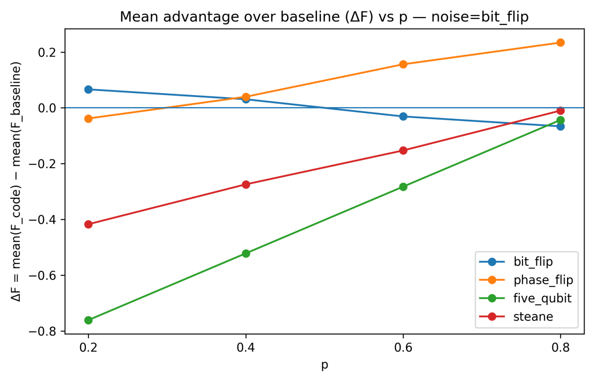
$$\mathcal{E}_Z(\rho) = (1-p)\rho + pZ\rho Z, \qquad K_0 = \sqrt{1-p}\,I, \quad K_1 = \sqrt{p}\,Z. \tag{17}$$

Depolarizing.

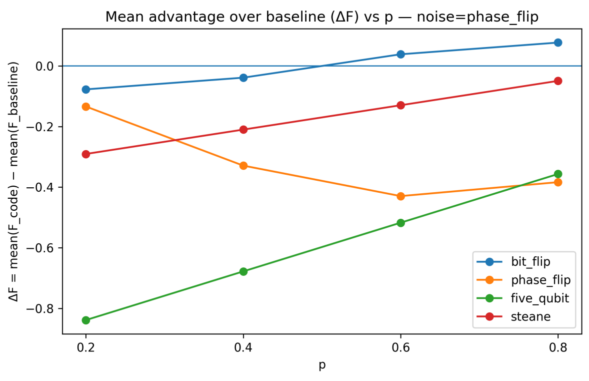
$$\mathcal{E}_{\text{dep}}(\rho) = (1-p)\rho + \frac{p}{3}\big(X\rho X + Y\rho Y + Z\rho Z\big), \tag{18}$$

with Kraus operators

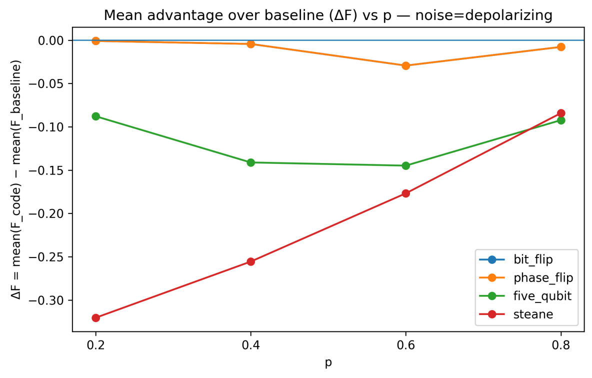
$$K_0 = \sqrt{1-p}\,I, \quad K_1 = \sqrt{\tfrac{p}{3}}\,X, \quad K_2 = \sqrt{\tfrac{p}{3}}\,Y, \quad K_3 = \sqrt{\tfrac{p}{3}}\,Z. \tag{19}$$



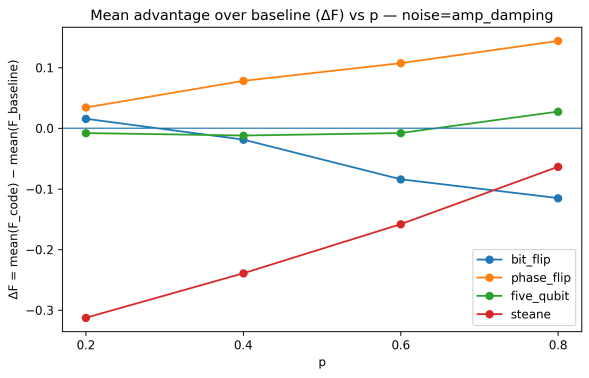
(a) Bit-flip noise



(b) Phase-flip noise



(c) Depolarizing noise



(d) Amplitude damping

Figure 1: Mean fidelity advantage over free-evolution baseline. Each panel shows $\Delta F(p) = \overline{F}_{\text{code}}(p) - \overline{F}_{\text{baseline}}(p)$ (averaged over all initial states) for a fixed noise model. Values above 0 indicate that the full encode–recover–decode pipeline improves performance relative to unencoded evolution for the same effective noise strength.

Amplitude damping.

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, \qquad K_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix} = \sqrt{p}\,|0\rangle\langle 1|\,. \tag{20}$$

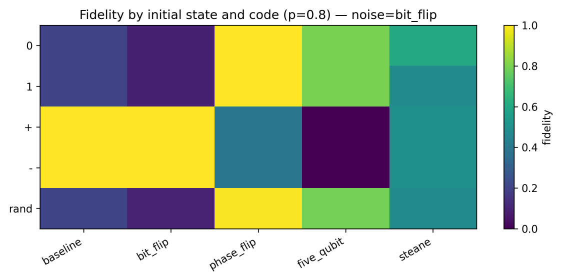
We validated encoding and decoding separately from noise by writing `test_codewords.py`, which checks that the encoders map $|0\rangle, |1\rangle$ to the intended logical codewords and that decoding inverts encoding on the codespace. We included the output in the Appendix.

For each choice of code, noise model, p, and input state, we executed the full circuit and recorded the resulting fidelity. We collected all fidelities into a single CSV for analysis and plotting. To manage runtime, the $[[5, 1, 3]]$ and Steane circuits used 10 shots per configuration. Our main entry point is `run_qec.py`, with noise channels in `noise_models.py` and code circuits in `qec_codes.py`. The full repository is linked in the Appendix.

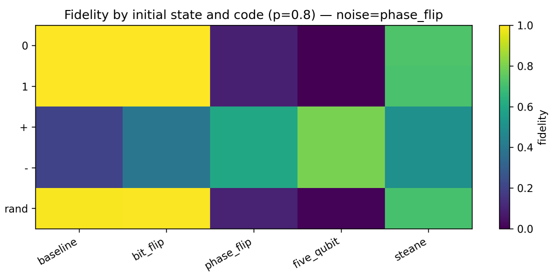
4 Results

We report performance using a mean advantage metric

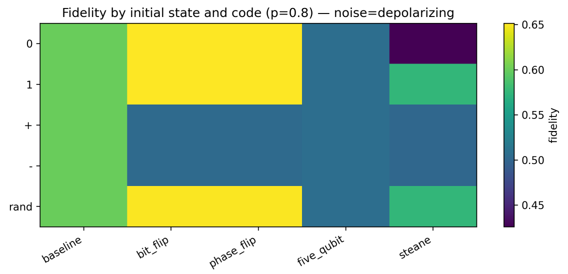
$$\Delta F(p) = \overline{F}_{\text{code}}(p) - \overline{F}_{\text{baseline}}(p), \tag{21}$$



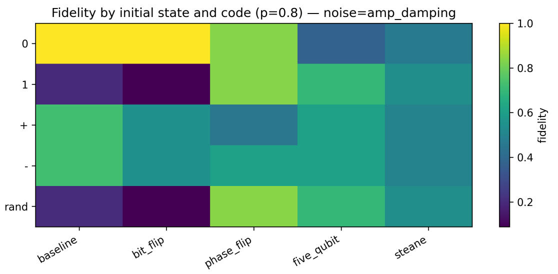
(a) Bit-flip noise



(b) Phase-flip noise



(c) Depolarizing noise



(d) Amplitude damping

Figure 2: State-dependence at fixed noise strength $p = 0.8$. Each heatmap reports the decoded fidelity by (initial state $\times$ code). This visualization exposes basis sensitivity and worst-case behavior that can be hidden by averaging.

where $\overline{F}$ denotes the mean fidelity over the five input states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle, a|0\rangle + b|1\rangle\}$ given a fixed noise channel and fixed error probability p. The baseline is the unencoded circuit that applies the same single qubit channel once and then measures fidelity. Each panel in Fig. 1 plots $\Delta F(p)$ as a function of p for one noise channel (averaged over choices of initial state): bit flip, phase flip, depolarizing, and amplitude damping.

Figure 1 shows that the 5 qubit and Steane codes give negative ΔF for most channels and most p values. The curves tend to move upward as p increases, but in the bit flip, phase flip, and depolarizing panels they stay below zero. So with this setup, the encoded pipeline does not beat the baseline on the mean over this input set.

The baseline depends on which basis the input state lives in, and this shows up clearly in the heatmaps. Figure 2 fixes $p = 0.8$ and plots fidelity by (input state $\times$ code) for each noise channel, one channel per quadrant. Under bit flip noise (panel a), the baseline keeps $|+\rangle$ and $|-\rangle$ near max fidelity while $|0\rangle$ and $|1\rangle$ drop a lot, and the bit flip code restores much of the fidelity for $|0\rangle$ and $|1\rangle$ under this channel. Under phase flip noise (panel b), the baseline keeps $|0\rangle$ and $|1\rangle$ high while $|+\rangle$ and $|-\rangle$ drop, and the phase flip code shows the expected protection for $|+\rangle$ and $|-\rangle$, though the effect is weaker than the bit flip case in panel (a). Depolarizing (panel c) distorts all states and the separation across inputs shrinks. Amplitude damping (panel d) produces a strong asymmetry between $|0\rangle$ and $|1\rangle$, consistent with relaxation.

Under phase flip noise, the heatmap shows that some encoded runs improve the X basis states relative to baseline at $p = 0.8$, yet ΔF stays near zero or negative for most codes. In Fig. 2(b), the baseline keeps $|0\rangle$ and $|1\rangle$ high and drives $|+\rangle$ and $|-\rangle$ down. The larger codes raise the X basis fidelities but they often reduce the Z basis fidelities.

Depolarizing noise compresses the differences. In Fig. 1(c), all curves sit near $\Delta F = 0$ or below, and the 5 qubit and Steane codes stay negative. In Fig. 2(c) at $p = 0.8$, fidelities cluster into a narrow band and still depend on the input basis. Here, encoding does not carve out a clear advantage.

Amplitude damping changes the ordering. In Fig. 1(d), the curve labeled `phase_flip` stays positive across p, the 5 qubit code stays near zero, and Steane stays negative but increases with p. In Fig. 2(d), the baseline shows a strong asymmetry between $|0\rangle$ and $|1\rangle$ at $p = 0.8$. The code dependence here looks tied to how the encoded state spreads $|1\rangle$ population over the block, rather than a clean single gate inversion picture.

5 Discussion

5.1 Interpretation

In this setup, the noise channel acts once after encoding and the rest of the circuit is ideal. So negative ΔF cannot come from noisy stabilizer measurements, noisy correction gates, or noisy decoding. It comes from the channel acting on an encoded state in a larger Hilbert space, and from the baseline including eigenstates of the channel that already keep high fidelity with no correction.

It is worth noting, ΔF is the difference between two means, taken over a fixed set of five input states, for one noise channel at a time. It is not a claim that a code with larger n wins. Since the input set includes channel eigenstates, the baseline can start close to fidelity one for part of the average. If a code improves the states that the channel actually damages, but it also reduces fidelity on states that were already stable, then the mean can still go down.

The heatmaps show the basis dependence that drives this. Under bit flip noise, the baseline keeps $|+\rangle$ and $|-\rangle$ high while $|0\rangle$ and $|1\rangle$ drop, and the bit flip code raises $|0\rangle$ and $|1\rangle$ back up. Under phase flip noise, the baseline keeps $|0\rangle$ and $|1\rangle$ high while $|+\rangle$ and $|-\rangle$ drop, and the phase flip code raises $|+\rangle$ and $|-\rangle$. The mean combines these behaviors, so it mixes a place where the baseline is already strong with a place where the baseline is weak. That mix is why ΔF can be negative even when protection exists on the vulnerable states.

The behavior of the 5 qubit and Steane codes also fits the noise model we chose. After encoding, we apply the channel independently to every physical qubit. For fixed p, increasing n increases the chance that more than one qubit is affected. A distance three code corrects one fault, but it does not correct typical weight two errors. At large p values, multi qubit errors become common, so the gap between a distance three promise and what the channel produces gets large. Here, negative mean advantage is not surprising even with ideal recovery and ideal decoding.

Amplitude damping needs separate treatment because it is not a flip channel. It does not act like applying one of $\{X, Y, Z\}$ with some probability. It moves population from $|1\rangle$ to $|0\rangle$ in a non unitary way. So the effect of encoding can depend on how the encoded state distributes $|1\rangle$ population across the block. This gives a different ordering than the Pauli channel panels, and it makes the results harder to summarize using the same language as bit flip or phase flip.

So the plots are telling two related things. The heatmaps show which specific states a code helps or hurts for a fixed channel and a fixed p. The ΔF curves compress that into one number per channel per p by averaging over a mixed input set that contains stable eigenstates. When those two views disagree, it is usually the average throwing away structure.

5.2 Limitations

We apply the same noise channel uniformly to every physical qubit. Real systems do not often behave like this. Different qubits may have different error rates, errors can drift over time, and error strength depends on the operation and on coupling.

We only apply noise once; we do not apply any noise during encoding, stabilizer measurement, conditional correction, or decoding. This massively simplifies the problem, whereas real world systems may not allow us to perform error correction steps under ideal conditions.

The 5 qubit and Steane results use 10 shots, so the fidelity estimates have large sampling error. This matters most when curves sit close together or when ΔF is near zero. We cannot confirm that the trends we observed can be generalized from such a small sample size.

The input set is small and structured. It is good to understand how these codes and noise channels affect common basis states, but it does not exhaust the Bloch sphere. So a mean over these five states is not a universal average fidelity, it is just a summary statistic for this set.

We only record decoded state fidelity. We do not log syndrome frequencies, which qubit was corrected, or whether multiple syndromes appear that should not. Without those logs, it is harder to separate a real code effect from a wrong syndrome map or an encoding mistake.

6 Conclusion

We implemented and tested four $[[n, 1, 3]]$ codes in a full correction pipeline and compared them to an unencoded baseline under bit flip, phase flip, depolarizing, and amplitude damping noise. For the parameter range tested, the larger codes often do not produce a positive mean advantage over baseline, even though they sometimes improve fidelity for the states that the channel degrades the most. The heatmaps make this visible: under bit flip noise the baseline preserves $|+\rangle$ and $|-\rangle$ while degrading $|0\rangle$ and $|1\rangle$; under phase flip noise the pattern is flipped.

References

[1] D. A. Lidar. *EE 514 Lecture Notes: Open Quantum Systems and Quantum Error Correction.*

[2] Qiskit Aer documentation, accessed December 2025.

[3] R. Lopez and D. Balasubramanian, "Final Project Proposal – EE 514," course document, 2025.

[4] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correcting code,"

[5] D. Gottesman, "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation," arXiv:0904.2557.

[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.

A Appendix

A.1 Repository

All code used to generate the circuits, run the simulator runs, and export the CSV is available below.

GitHub: `https://github.com/balabala67/PHYS-514-Final-Project`

A.2 Test Script Output

We include the full outputs of the test scripts as separate files and link them below.

- `test_noise_models.py` **output:** `https://docs.google.com/document/d/1z3DYAWqboOFlLOwJQxHYlWoc2J` `W9E/edit?usp=sharing`
- `test_codewords.py` **output:** `[https://docs.google.com/document/d/1Si-lSe4rnhwBBmA5Xap5XjndBEKU` `edit?usp=sharing`