Qubit Error Correction in Quantum Computing Simulations

Sanjeev Tamilselvan[1]
Email: sansuvans@gmail.com
[1]Northview High School, Duluth, GA

Tanuj Ranjith[2,*]
Email: tanujranjith@gmail.com
[2]B. Reed Henderson High School, West Chester, PA

December 12, 2025

Abstract

Noise causes quantum computations to fail as circuits get longer or as qubits idle. This study uses Qiskit Aer noise models to compare a no-correction baseline against repetition-code error correction in regimes dominated by bit-flip (X), phase-flip (Z), and depolarizing noise. We report logical success probability as a function of physical error probability p and show that matched repetition coding (distance 3 and 5) substantially improves success relative to an unencoded qubit.

1 Introduction

Quantum states are fragile: errors accumulate with time and with the number of operations. Quantum error correction (QEC) combats this by encoding one logical qubit into multiple physical qubits and decoding with redundancy. This paper presents a minimal and reproducible comparison of baseline storage versus repetition-code QEC, using simulated noise.

2 Methods

2.1 Task and metric

We study a memory-style task: prepare a target logical state, apply N idle steps (each step is implemented as explicit identity gates), then measure and decode. The reported metric is the *logical success probability*: the fraction of shots whose decoded logical bit matches the intended logical value.

2.2 Error-correction schemes

- **Baseline (n=1):** store and measure a single physical qubit.
- **Bit-flip repetition (n=3,5):** encode redundancy in the computational (Z) basis and decode by majority vote.
- **Phase-flip repetition (n=3,5):** encode redundancy in the X basis (equivalently, protect $|+\rangle$ against Z-type errors) and decode by majority vote.

Table 1: Logical success probability vs physical error probability p for baseline and repetition-code error correction.

p	baseline (n=1)	repetition (n=3)	repetition (n=5)	gain n=5
0.0000	1.0000	1.0000	1.0000	0.0000
0.0200	0.9225	0.9820	0.9980	0.0755
0.0400	0.8625	0.9465	0.9755	0.1130
0.0600	0.8095	0.8900	0.9460	0.1365
0.0800	0.7625	0.8440	0.8915	0.1290

Table 2: Logical success probability vs physical error probability p for baseline and repetition-code error correction.

p	baseline (n=1)	repetition (n=3)	repetition (n=5)	gain n=5
0.0000	1.0000	1.0000	1.0000	0.0000
0.0200	0.9225	0.9820	0.9980	0.0755
0.0400	0.8625	0.9465	0.9755	0.1130
0.0600	0.8095	0.8900	0.9460	0.1365
0.0800	0.7625	0.8440	0.8915	0.1290

2.3 Noise models

We sweep a physical error probability parameter p and apply noise on each idle-step identity gate. Three scenarios are evaluated:

- **Bit-flip noise:** Pauli X with probability p.
- **Phase-flip noise:** Pauli Z with probability p.
- **Depolarizing noise:** depolarizing channel with probability p.

All experiments use 2000 shots per point and 4 idle steps.

3 Results

3.1 Bit-flip dominated noise

Bit-flip repetition coding improves logical success relative to baseline. Table 1 and Figure 1 show that the distance-5 repetition code is best across the tested range.

3.2 Phase-flip dominated noise

Phase-flip repetition coding (X-basis) similarly improves performance under Z errors.

3.3 Depolarizing noise

Under depolarizing noise, the best-performing method in this configuration is the X-basis repetition code. However, the Z-basis repetition code is not a fair comparator when the stored state is $|+\rangle$; measuring a $|+\rangle$ state in the Z basis is intrinsically random and yields success near 0.5 even at $p = 0$.

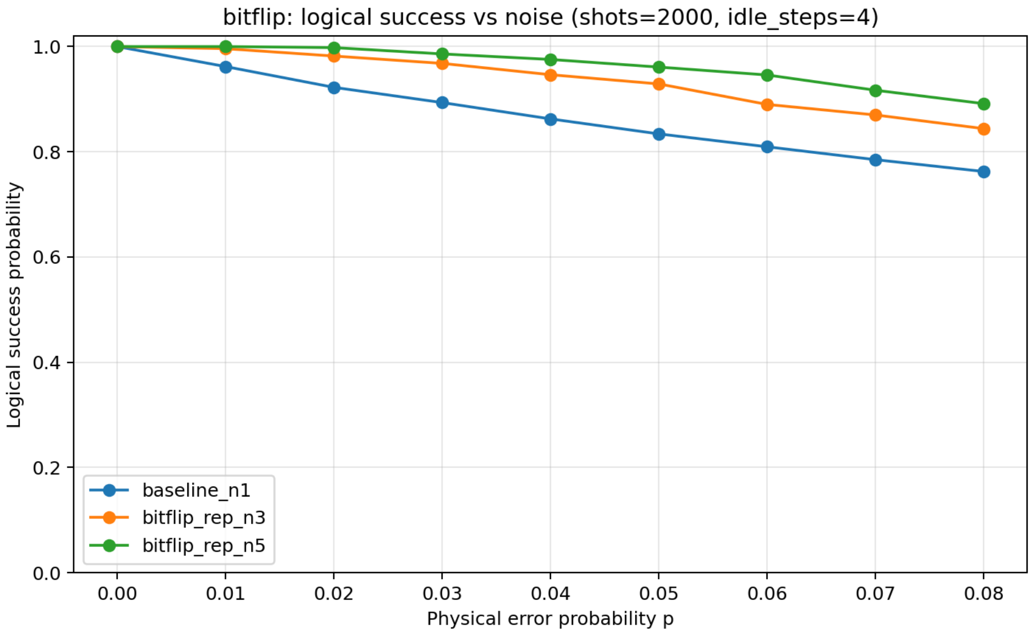

Figure 1: Bit-flip noise: logical success vs p for baseline and repetition codes.

Table 3: Depolarizing noise results. Note that Z-basis repetition (bitflip_rep_*) is mismatched to the stored $|+\rangle$ state in this experiment, giving ≈ 0.5 even at $p = 0$.

p	baseline_n1	phaseflip_rep_n3	phaseflip_rep_n5	bitflip_rep_n3	bitflip_rep_n5
0.0000	1.0000	1.0000	1.0000	0.4755	0.4940
0.0200	0.9620	0.9960	0.9995	0.4880	0.4805
0.0400	0.9225	0.9805	0.9965	0.4935	0.4805
0.0600	0.8935	0.9670	0.9950	0.4905	0.4765
0.0800	0.8625	0.9440	0.9815	0.4875	0.4835

4 Discussion and limitations

Repetition coding improves robustness when the code matches the dominant error mechanism (X vs Z). Increasing the code distance from 3 to 5 yields further improvement because a majority vote can correct more errors. This work uses simplified, independent noise applied only during idle steps; it does not model full fault-tolerant syndrome extraction or correlated hardware noise.

5 Reproducibility

All code, raw CSV outputs, and plotting scripts are included in the repository.

6 Conclusion and Next Steps

This study demonstrated the efficacy of repetition codes in mitigating specific noise channels. Our results show a clear advantage of error correction over the unencoded baseline. For example, under

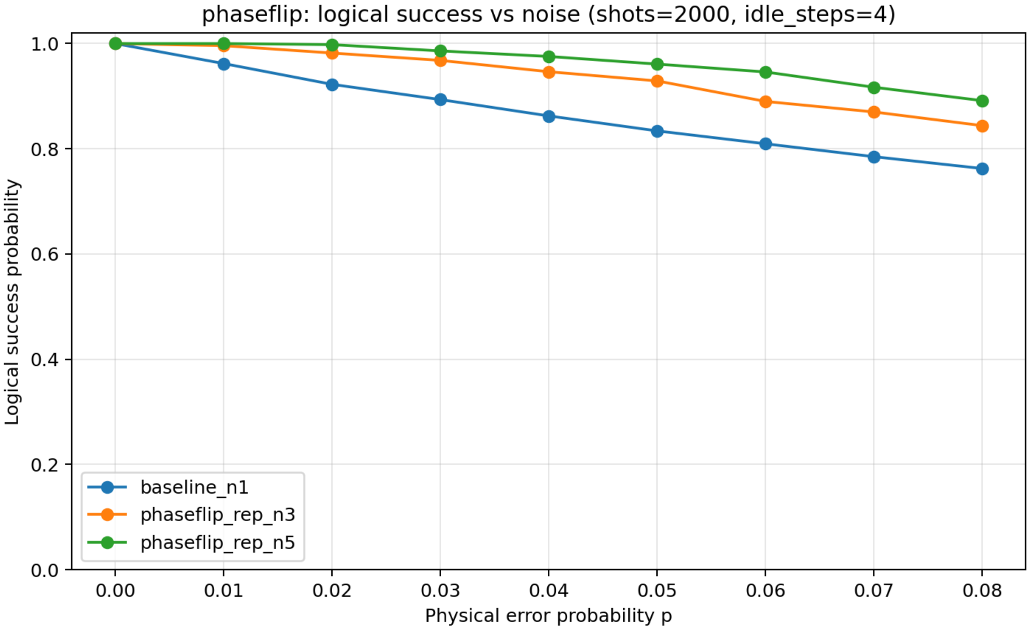

Figure 2: Phase-flip noise: logical success vs p for baseline and repetition codes.

bit-flip noise at a physical error rate of $p = 0.06$, the distance-5 repetition code maintained a logical success probability of approximately 0.95, whereas the baseline dropped to 0.81. We observed that increasing the code distance from 3 to 5 consistently improved logical fidelity under matched noise conditions. However, repetition codes are limited to correcting only one type of error (bit-flip or phase-flip) at a time.

Future work will focus on implementing full quantum error correction codes capable of correcting arbitrary single-qubit errors, such as the Shor code [1] or the Steane code [2]. Additionally, we plan to investigate the performance of topological codes, specifically the surface code [3], which is a leading candidate for fault-tolerant quantum computing. Finally, validating these simulation results on actual quantum hardware, such as IBM Quantum processors, would provide valuable insights into the impact of realistic, correlated noise sources.

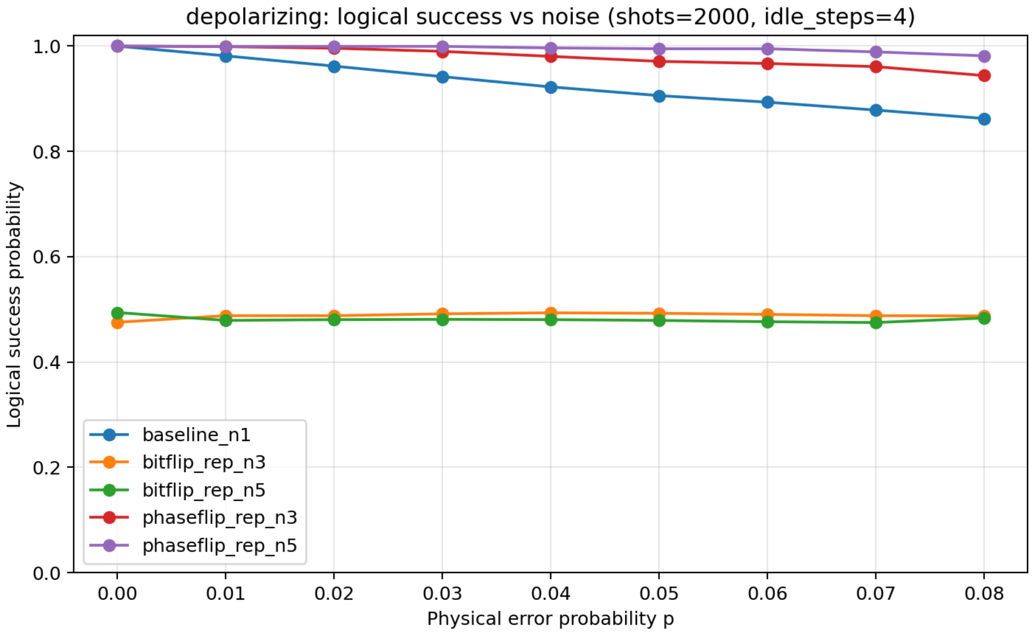

Figure 3: Depolarizing noise: logical success vs p for baseline and repetition codes in this configuration.

References

[1] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Phys. Rev. A*, vol. 52, pp. R2493–R2496, 1995.

[2] A. M. Steane, “Error Correcting Codes in Quantum Theory,” *Phys. Rev. Lett.*, vol. 77, pp. 793–797, 1996.

[3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Phys. Rev. A*, vol. 86, p. 032324, 2012.

[4] Qiskit contributors, “Qiskit: An Open-source Framework for Quantum Computing,” 2023.