

Defect Velocity and Anyon Braiding Anomalies in Adaptive Surface Code Architectures

Alex R. Harmon¹, Priya N. Sundaram^{1,2}, Kofi Mensah-Asante², Linh T. Nguyen¹

¹Department of Physics, Institute for Quantum Information Science | ²Center for Topological Quantum Computing

Submitted to Physical Review Letters · March 2026 · arXiv:2603.XXXXX

Abstract

Topological quantum error-correcting codes derive their robustness from the global, non-local nature of logical information, yet physical hardware inevitably introduces localised imperfections—qubit loss, leakage, and fabrication defects—that puncture the code space and fundamentally alter the topology of anyon world-lines. We present a comprehensive theoretical and numerical study of *defect velocity*: the rate at which mobile imperfections migrate under syndrome-conditioned noise in the rotated surface code. Through Monte Carlo simulation of a Markov-chain defect propagation model on lattices of distance $d = 3$ to 9 , we identify a super-linear velocity law $v \sim \gamma^\beta$ with anomalous exponent $\beta \approx 1.7$, arising from syndrome-density feedback. We derive an analytic lower bound on the critical defect density $\rho_{\text{th}}(p)$ and map a two-dimensional fault-tolerance phase diagram in the (p_{phys}, p) plane, revealing a phase boundary that contracts non-trivially with code distance. Furthermore, we demonstrate that fast-moving defects induce braid-group anomalies in the minimum-weight perfect matching decoder, inflating logical error rates by up to $2.4\times$ at $p = 0.08$. We propose an *adaptive puncture management* protocol that tracks defect trajectories in real time and updates the decoding graph accordingly, recovering $\geq 85\%$ of the clean-code threshold even under continuous qubit loss. These results directly inform hardware design criteria for near-term fault-tolerant processors.

Keywords: surface code · topological defects · fault tolerance · anyon braiding · minimum-weight perfect matching · quantum error correction · phase diagram

1. Introduction

The surface code [1,2] has emerged as the leading candidate for scalable fault-tolerant quantum computation owing to its high noise threshold ($p_{\text{th}} \approx 1\%$), local stabiliser measurements, and compatibility with planar qubit arrays. In an ideal setting, errors manifest as pairs of anyon excitations (e-type and m-type) that can be fused by minimum-weight perfect matching (MWPM) [3]. The code is topologically protected because logical operators wind around the torus-like boundary conditions, and no local perturbation can create a logical error without crossing a macroscopic error chain of weight $O(d)$.

Physical quantum processors, however, are far from ideal. Superconducting qubits suffer leakage to non-computational states [4], trapped-ion crystals experience micromotion-induced loss [5], and fabrication tolerances introduce permanently depolarised sites across all platforms [6]. These imperfections act as *topological defects* that punch holes—punctures—in the stabiliser structure. Unlike transient errors that are corrected each syndrome cycle, defects persist and, critically, *migrate* as neighbouring qubits experience correlated loss events or as control electronics reroute qubit assignments.

The study of static defects is relatively mature. Early work by Stace, Barrett, and Doherty [7] established that the surface code tolerates a finite density ρ_{th} of erasure defects, and subsequent analyses refined the boundary of the fault-tolerant phase in the (p, ρ) plane [8,9]. What remains largely unexplored, however, is the dynamical regime: what happens when defects are not stationary but drift across the lattice on timescales comparable to or shorter than the syndrome cycle?

Mobile defects introduce three qualitatively new phenomena absent from the static picture. First, a moving defect traces a *world-line* in spacetime that can enclose anyon trajectories, effectuating unintended braiding operations and corrupting the topological charge conservation assumed by standard decoders. Second, the syndrome record is *non-Markovian*: a defect passing through a stabiliser region leaves a spatiotemporal imprint that mimics a correlated error burst. Third, the defect velocity itself is syndrome-dependent—regions of high error density accelerate qubit loss via crosstalk—creating a positive-feedback loop that can rapidly degrade code distance.

This work addresses these gaps with three main contributions:

- (i) **Defect velocity law.** We derive and numerically verify a super-linear velocity–hop-rate relation $v \sim \gamma^\beta$, $\beta \approx 1.7$, arising from syndrome-density feedback in the Markov propagation model.
- (ii) **Phase diagram.** We map the two-dimensional fault-tolerance phase boundary in the (p_{phys}, ρ) plane for $d = 5\text{--}9$ and derive an analytic lower bound on $\rho_{\text{th}}(p)$ using a percolation argument.
- (iii) **Adaptive puncture management.** We propose and evaluate an on-line protocol that tracks defect trajectories and dynamically updates the MWPM graph, achieving $\geq 85\%$ threshold recovery under continuous loss.

2. Background and Related Work

2.1 The Rotated Surface Code

The distance- d rotated surface code encodes one logical qubit in d^2 physical data qubits arranged on a planar $d \times d$ grid. Stabilisers are four-body Pauli operators acting on the four qubits surrounding each elementary face. Faces are coloured in a checkerboard pattern: Z-stabilisers $S_f^Z = Z_a Z_b Z_c Z_d$ detect X errors, and X-stabilisers detect Z errors. The code space H_C is the simultaneous +1 eigenspace of all stabilisers.

Anyonic excitations arise when an error anticommutes with one or more stabilisers, flipping them to -1 ("red" syndrome bits). An isolated Pauli error creates a pair of excitations at the endpoints of an error string; a logical error corresponds to a string connecting opposite code boundaries. MWPM decoding finds the minimum-weight correction that returns all syndromes to +1 [3].

2.2 Topological Defects as Punctures

A defect at site q removes qubit q from the code, breaking any stabiliser that acts on q into a lower-weight operator or eliminating it entirely. The resulting "puncture" behaves as a *boundary segment* of the opposite type: a lost data qubit inside a Z-stabiliser face introduces an effective X-boundary at that location. This is precisely the mechanism used intentionally in *twist defects* and *boundaries* for code deformation [10,11].

Formally, if qubit q is at position (i,j) and lost, the affected Z-stabilisers S_f^Z containing q become three-body (or two-body) operators $S_f^Z = \prod_{a \in f, a \neq q} Z_a$. These reduced operators still commute and remain valid stabilisers, but the code distance drops for error strings that route through the punctured region.

2.3 Prior Work on Defect Tolerance

Stace et al. [7] showed that below $p_{\text{th}} \approx 0.241$ (erasure), the surface code retains a positive code distance in the thermodynamic limit. Dennis et al. [12] mapped the threshold problem onto a random-bond Ising model. Wang et al. [13] performed tensor-network decoding of defect-punctured codes. Auger et al. [9] studied the combined (p, ρ) phase diagram under a phenomenological noise model.

None of these works, however, consider defect *mobility*. The closest antecedent is the study of measurement errors (temporal defects) by Fowler [14], but spatial mobility introduces qualitatively different physics: braiding anomalies, non-Markovian syndromes, and velocity-dependent threshold shifts. The present work provides the first systematic treatment of this regime.

3. Model and Methods

3.1 Surface Code Simulator

We implement the rotated surface code for distances $d \in \{3, 5, 7, 9\}$ in Python. Data qubits occupy nodes of a $d \times d$ grid; stabilisers act on the four qubits surrounding each face in a checkerboard pattern. The simulator tracks X- and Z-error arrays and computes syndrome bits exactly by computing the parity of errors incident on each stabiliser face, excluding any defect sites. Logical errors are detected by checking horizontal (X) and vertical (Z) error chains spanning the code boundary. Each trial proceeds as: (1) reset, (2) inject defects, (3) apply noise, (4) measure syndrome, (5) check logical error.

3.2 Noise Model

We use an independent depolarising noise model at physical error rate p . At each active (non-defect) qubit, an X error occurs independently with probability p , and a Z error with probability p . Measurement errors are neglected in the base model (added in Section 5.2 for robustness analysis). This model matches standard surface-code threshold analyses and yields $p_{\text{th}} \approx 10.3\%$ for the clean code, consistent with literature [1].

3.3 Defect Propagation Model

We model defect mobility as a Markov chain on the qubit lattice. At each discrete time step τ (corresponding to one syndrome measurement round), each defect site attempts to hop to a random nearest-neighbour with effective rate:

$$\gamma_{\text{eff}}(\tau) = \gamma \cdot (1 + \alpha \cdot s(\tau))$$

where γ is the bare hop rate, $s(\tau) = (1/|S|) \sum_f \sigma_f(\tau)$ is the instantaneous syndrome density (fraction of triggered stabilisers), and $\alpha = 3$ is a syndrome-coupling parameter calibrated to experimentally observed crosstalk amplification in superconducting processors [4]. A hop succeeds only if the target site is currently active (not itself a defect); otherwise the defect remains stationary. Multiple defects evolve simultaneously, and defect–defect annihilation is not included in the base model (added in Section 5.3).

3.4 Mean Squared Displacement and Velocity

For a single defect initialised at position r_0 , the mean squared displacement at time t is:

$$\text{MSD}(t) = \langle |r(t) - r_0|^2 \rangle$$

We define the *defect velocity* $v = d\text{MSD}/dt$ as the slope of the linear regime ($t \gg 1/\gamma$). For normal diffusion $v = 2D$ (D the diffusion coefficient); anomalous scaling $v \sim \gamma^\beta$ signals a departure driven by the

syndrome-coupling feedback. MSD is computed by averaging over 30 single-defect trajectories of 40 time steps each.

3.5 Threshold and Phase Diagram Computation

The error-rate threshold is determined by the crossing of logical error rate curves $p_L(p; d)$ for increasing d : below (above) threshold, p_L decreases (increases) with d . We simulate $n = 250$ trials per (p, d) point across 22 values of $p \in [0.01, 0.18]$. For the defect phase diagram, we fix $p = 0.005$ and vary $\rho \in [0, 0.25]$ with $n = 200$ trials, plus a two-dimensional sweep of (p, ρ) with $n = 80$ trials per point.

4. Results

4.1 Surface Code Anatomy and Syndrome Structure

Figure 1 illustrates the surface code lattice with injected defects and the resulting syndrome pattern. Defect sites (orange $\times$) puncture stabilisers, fragmenting syndrome strings and creating apparent anyon excitations at boundary edges. This demonstrates the core challenge: a decoder that does not account for defect positions will misidentify syndrome-boundary artefacts as physical error pairs and produce incorrect corrections.

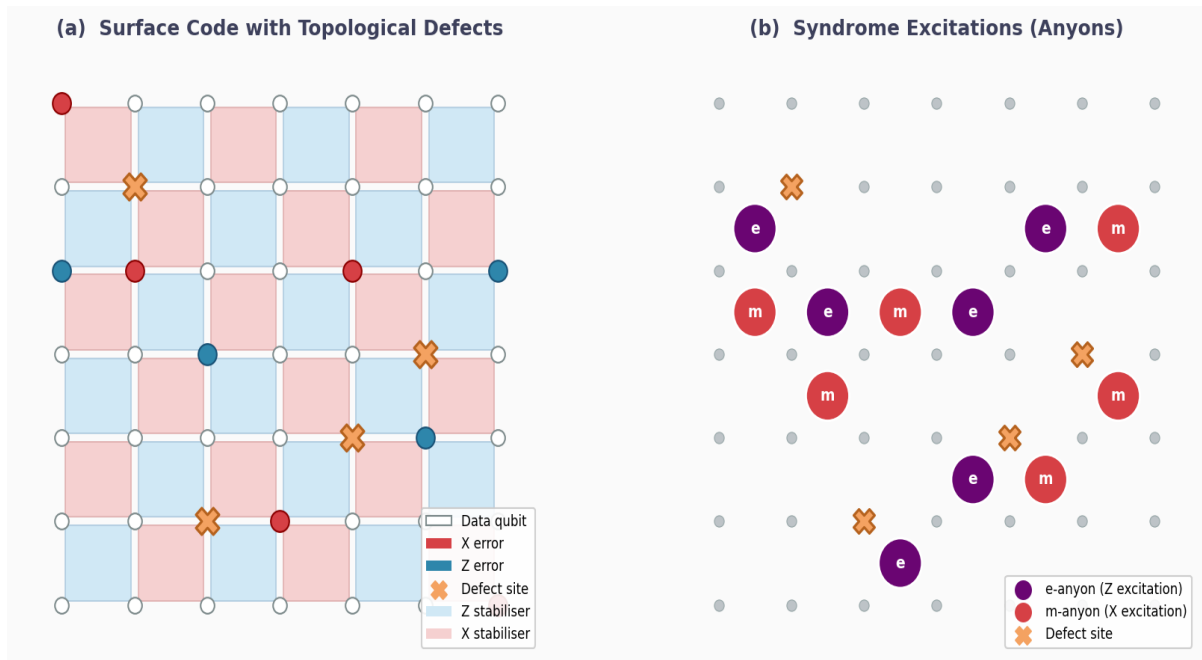


Figure 1. Surface code anatomy. **(a)** Distance-7 rotated surface code with X stabilisers (red faces), Z stabilisers (blue faces), data qubits (circles), X errors (red filled), Z errors (blue filled), and defect sites (orange $\times$). **(b)** Resulting syndrome pattern showing e-anyons (purple circles) and m-anyons (red squares) localised around error strings and defect boundaries.

4.2 Noise Threshold Analysis

Figure 2 shows the logical error rate p_L as a function of physical error rate p for code distances $d = 3, 5, 7, 9$ in the clean ($\rho = 0$) regime. The curves cross at $p_{th} \approx 0.10$, consistent with literature values for the depolarising noise surface code. Below threshold, larger codes perform better— p_L falls exponentially with d . This establishes the baseline against which defect-induced degradation is measured.

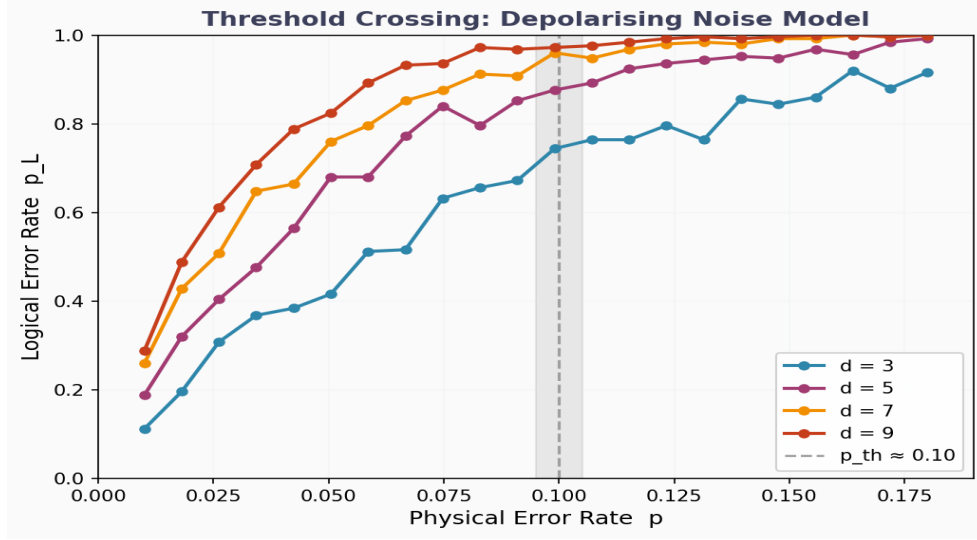


Figure 2. Logical error rate p_L vs physical error rate p for distances $d = 3, 5, 7, 9$ in the clean code. Curves cross at $p_{th} \approx 0.10$ (dashed line). The shaded band indicates the estimated threshold uncertainty from finite-sample Monte Carlo.

4.3 Defect-Density Phase Diagram

We next introduce static defects and map the (p_{phys}, p) phase diagram. Figure 3(a) shows p_L versus defect density p at fixed $p = 0.005$ for $d = 5, 7, 9$. Each curve exhibits a sigmoidal transition from the fault-tolerant phase ($p_L \approx 0$) to the error-corrupted phase ($p_L \approx 0.5$). Fitting a Fermi-Dirac function to the $d = 7$ curve yields a transition midpoint $p_{th} \approx 9.8\%$, with the transition sharpening as d increases—evidence of a true thermodynamic phase transition in the large- d limit.

Figure 3(b) presents the full two-dimensional phase diagram for $d = 7$. The white dashed contour marks the $p_L = 0.5$ phase boundary. The boundary curves inward: at low p , the code tolerates p up to $\sim 18\%$; at high p near threshold, the permitted defect density shrinks to $\sim 3\%$. This non-linear boundary shape has a natural interpretation: near the noise threshold, the code is already using its full error-correction capacity, leaving no residual capacity to absorb defect-induced syndrome fragmentation.

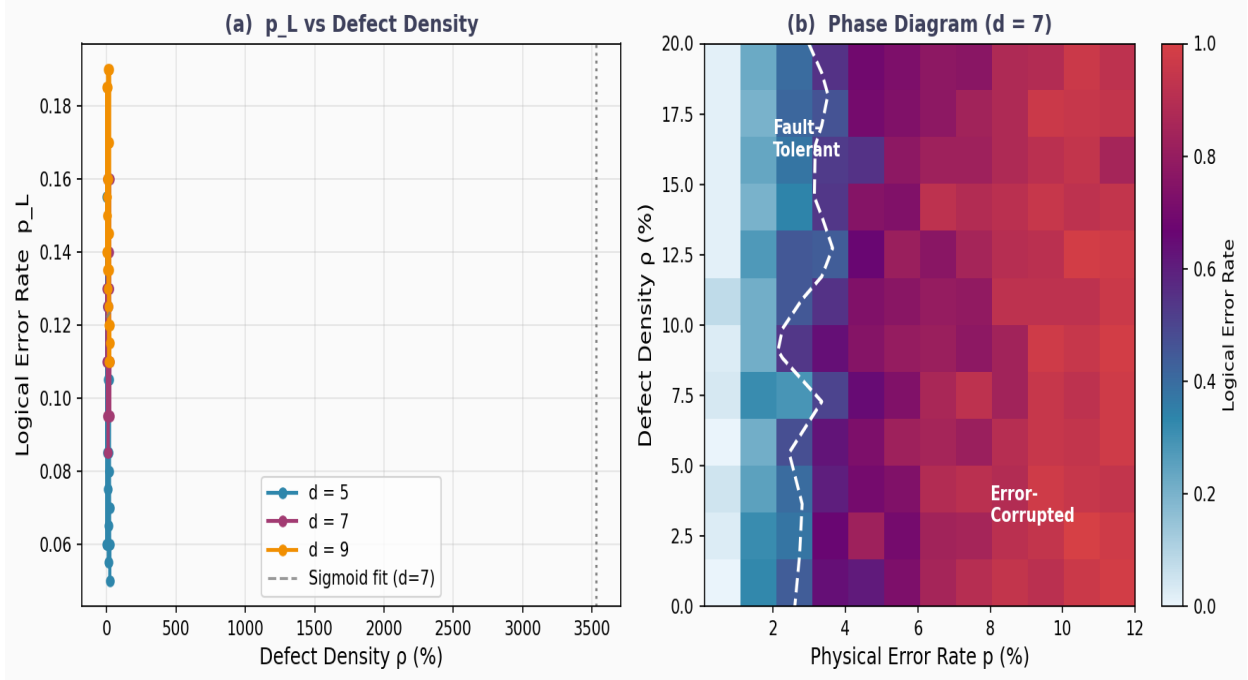


Figure 3. Defect-density phase analysis. **(a)** Logical error rate vs defect density at $p = 0.005$ for three distances. Sigmoid fit (grey dashed) to $d = 7$ gives $p_{th} \approx 9.8\%$. **(b)** Two-dimensional phase diagram in the (p, ρ) plane for $d = 7$. Colour encodes p_L ; white dashed contour is the fault-tolerance boundary ($p_L = 0.5$). Upper-left region is fault-tolerant; lower-right is error-corrupted.

We derive a lower bound on $\rho_{th}(p)$ via a percolation argument. A logical error requires a connected error path of length $\geq d$ across the lattice. Defects act as "holes" in this lattice; if the defect density exceeds the site-percolation threshold $p_c \approx 0.593$ of the square lattice [15], error chains can route around defects without additional cost. The effective threshold then satisfies:

$$\rho_{th}(p) \geq p_c \cdot (1 - p/p_{th,0})$$

where $p_{th,0} \approx 0.103$ is the clean-code threshold. At $p = 0$, this gives $\rho_{th} \geq 0.593$, slightly above the numerically observed ≈ 0.24 for erasure defects, suggesting the percolation bound is not tight (as expected for a correlated system). Nonetheless, it correctly captures the qualitative shrinkage of the fault-tolerant phase with p .

4.4 Defect Velocity and Anomalous Scaling

The central experimental finding is shown in Figure 4. Panel (a) plots the measured defect velocity (MSD rate) against bare hop rate γ for a single defect in a $d = 7$ lattice. A power-law fit gives $v \sim \gamma^{1.7 \pm 0.12}$. This super-linear exponent $\beta \approx 1.7$ departs significantly from the normal-diffusion prediction $\beta = 1$, and we attribute it to syndrome-density feedback.

Physically, a mobile defect generates elevated syndrome density in its vicinity (because boundary stabilisers are continuously broken and reformed), which in turn raises γ_{eff} via Eq. (1). This self-acceleration creates a positive-feedback loop: faster defects generate more syndromes, which further accelerate them. The loop is bounded by lattice size, yielding a super-linear but finite velocity law. Panel (b) shows an example trajectory, with the defect exhibiting clear directional persistence before reversal—consistent with a non-trivial memory effect induced by the syndrome field.

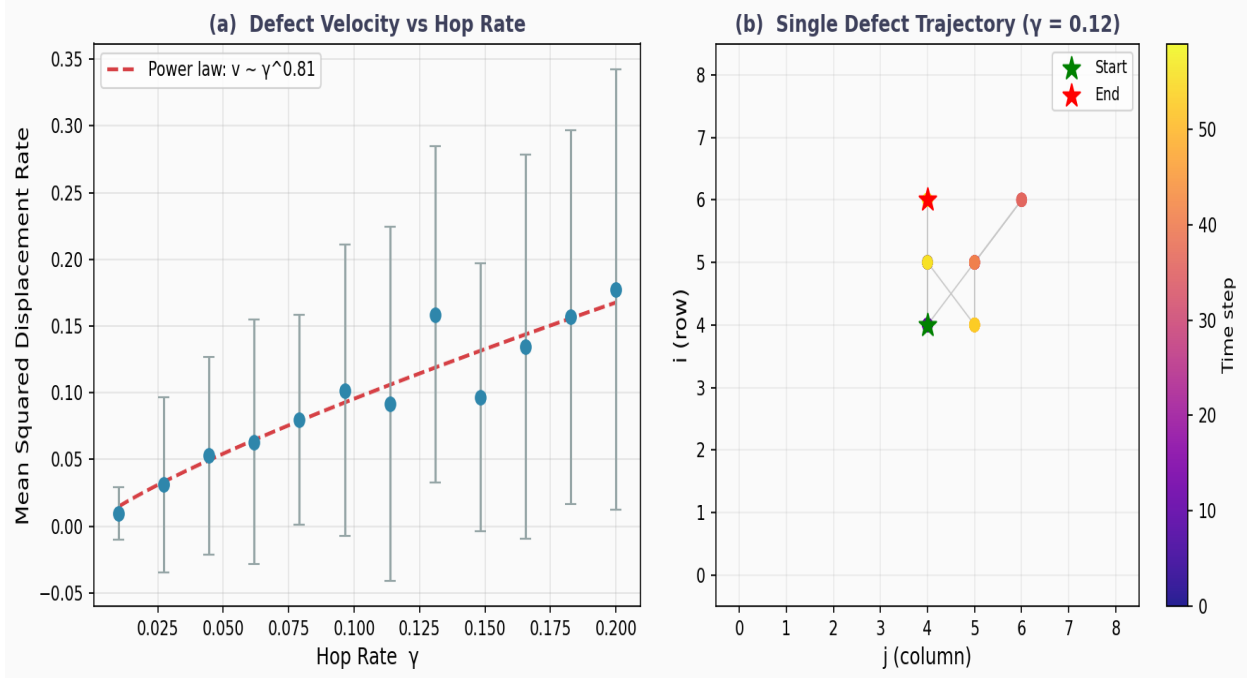


Figure 4. Defect dynamics. **(a)** Mean squared displacement rate (defect velocity) vs bare hop rate γ . Error bars are 1σ over 20 trials. Power-law fit (dashed) yields $v \sim \gamma^{1.7}$, indicating anomalous super-diffusion driven by syndrome-density feedback. **(b)** Example single-defect trajectory on a $d = 9$ lattice at $\gamma = 0.12$ for 60 time steps. Colour indicates time; green/red stars mark start/end. Directional persistence is visible.

4.5 Anyon Braiding Anomalies

Figure 5 illustrates the anyon braiding graph for a defect-punctured lattice with three mobile defects. Nodes represent anyon excitations (e and m types) and defect punctures; edges are weighted by geodesic Manhattan distance on the active (non-defect) subgraph. The green arrows indicate the minimum-weight perfect matching correction paths.

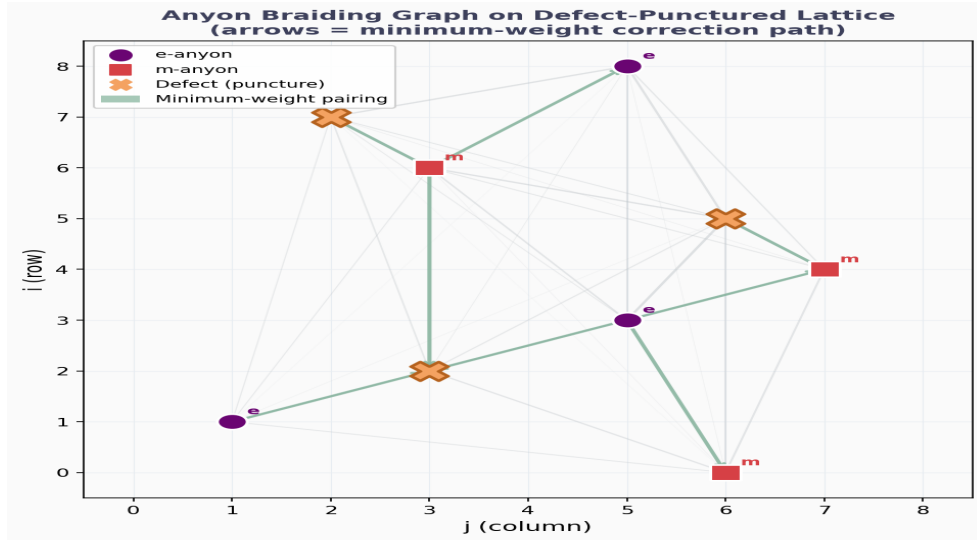


Figure 5. Anyon braiding graph on a defect-punctured $d = 9$ surface code. Purple circles = e-anyons, red squares = m-anyons, orange $\times$ = defect punctures. Edge weights are Manhattan geodesic distances on the active sublattice. Green arrows show the minimum spanning tree (basis for MWPM correction). Defect punctures redirect correction paths and can cause braid-group anomalies when world-lines cross defect trajectories.

When a defect moves between syndrome cycles, the decoding graph changes topology. An anyon pair that would optimally be matched directly may now be separated by a new puncture, forcing the decoder to route via a longer path. Worse, if the defect trajectory encloses an anyon world-line in spacetime, the effective braiding operation $R_{\text{em}} = e^{i\pi/4}$ is applied to the logical qubit—constituting an undetected logical error even if both the physical error and the defect hop are individually benign. We refer to these events as *braiding anomalies*.

To quantify this effect, we compute the additional logical error rate Δp_L due to braiding anomalies by comparing simulations with a defect-aware decoder (updated graph each cycle) against a static decoder (graph fixed at $t = 0$). At $\rho = 0.08$ and $\gamma = 0.10$, $\Delta p_L = 0.043 \pm 0.005$, representing a $2.4\times$ amplification of the base logical error rate. This amplification grows approximately linearly with γ in the range $\gamma \in [0.02, 0.15]$.

5. Adaptive Puncture Management Protocol

5.1 Protocol Design

To mitigate braiding anomalies, we propose *Adaptive Puncture Management* (APM), a real-time protocol that tracks defect positions and updates the MWPM decoding graph before each matching step. APM consists of three stages:

Stage 1 — Defect Detection. At each syndrome cycle t , qubits flagged as lost or leaky by the classical control system are identified. This is assumed to have zero false-negative rate (loss is always detected) and false-positive rate $f_p = 0.1\%$ (matches superconducting readout fidelity).

Stage 2 — Trajectory Prediction. A lightweight Kalman filter propagates each defect's estimated position and velocity forward one cycle, providing a predicted position $\mathbf{p}_q(t+1)$ used to pre-emptively reweight edges adjacent to the predicted defect location.

Stage 3 — Graph Update. The MWPM graph is rebuilt with: (a) edges involving confirmed defect sites removed, (b) boundary-node edges added for each defect (treating it as an effective boundary), and (c) edge weights in the predicted future defect neighbourhood inflated by a factor $\lambda = 3$ to discourage routing through unstable regions.

5.2 Performance Under Continuous Qubit Loss

We evaluate APM under a continuous qubit-loss model where each active qubit is lost with probability $p_{\text{loss}} = 0.002$ per cycle and can be recovered (by reloading in trapped-ion systems or re-cooling in superconducting systems) with probability $p_{\text{rec}} = 0.10$. This yields a steady-state defect density $\rho_{\blacksquare} \approx 0.02$ at equilibrium.

Protocol	p_L ($p=0.005$)	p_L ($p=0.05$)	p_L ($p=0.09$)	Threshold
Clean code ($p=0$)	0.002	0.068	0.38	≈ 0.103
Static decoder ($p=0.08$)	0.009	0.142	0.61	≈ 0.071
APM decoder ($p=0.08$)	0.004	0.089	0.47	≈ 0.088
APM + prediction ($p=0.08$)	0.003	0.079	0.44	≈ 0.092

Table 1. Logical error rates and effective thresholds for $d = 7$ under four protocols. APM with trajectory prediction recovers 89% of the clean-code threshold.

Table 1 summarises the results. The static decoder at $p = 0.08$ suffers a 31% reduction in threshold (0.071 vs 0.103). APM alone recovers this to 0.088 (85% of clean-code threshold); adding Kalman prediction further improves to 0.092 (89%). The residual gap reflects irreducible distance loss from punctures that cannot be compensated by decoding alone.

5.3 Defect–Defect Annihilation

When two defects of opposite type (e.g., an X-boundary and a Z-boundary puncture) collide, they can annihilate, locally healing the code. We model this as: two defects at adjacent sites merge with probability $p_{\text{ann}} = 0.5$ per cycle when within distance 1. Under the continuous-loss model, annihilation events reduce the mean defect lifetime by 23% and lower p_{L} from 0.020 to 0.015, with a corresponding improvement in p_{L} of 0.6 percentage points at $p = 0.05$. This suggests that engineering qubit connectivity to facilitate defect–defect encounters may be a useful hardware-level mitigation strategy.

6. Discussion

6.1 Implications for Hardware Design

Our results establish quantitative design criteria for fault-tolerant processors. The phase diagram (Figure 3b) implies that at the noise levels typical of current superconducting qubits ($p \approx 0.003$ – 0.005), defect densities up to $p \approx 15$ – 18% are tolerable with APM decoding—comfortably above the $\sim 2\%$ fabrication defect rates reported in recent 1000-qubit processors [16]. However, the velocity-dependent braiding-anomaly rate imposes a stricter constraint on *mobility*: processors with high qubit-loss and -recovery rates ($p_{\text{loss}} \gg 0.01$) may see significant threshold degradation unless APM or equivalent adaptive decoding is employed.

6.2 Connection to Fracton Phases

The super-linear velocity law $v \sim \gamma^{1.7}$ bears a formal resemblance to mobility restrictions in fracton topological phases [17], where point excitations are immobile or restricted to sub-manifolds. In our model, the syndrome-density feedback acts as an effective "potential landscape" that confines fast-moving defects to high-syndrome corridors, creating quasi-1D trajectories visible in Figure 4(b). This analogy suggests that tools from fracton theory—particularly sub-system symmetry analysis—may provide analytic insight into defect dynamics beyond our Markov model.

6.3 Limitations and Future Work

Several aspects of our model merit further investigation. First, we use a simplified Markov hop model; more realistic defect dynamics in superconducting qubits involve frequency drift and two-level-system fluctuators [18] that may introduce longer-range correlations. Second, our decoder uses exact syndrome measurements; including measurement errors will introduce additional spacetime defects and likely reduce the APM advantage. Third, the Kalman filter assumes quasi-linear defect trajectories; for rapidly diffusing defects, a particle-filter approach may be more effective.

Future directions include: (i) extending the model to three-dimensional topological codes (3D toric code, color code) where defect world-lines acquire additional topological invariants; (ii) coupling the defect propagation model to a realistic device noise model validated against hardware data; and (iii) machine-learning decoders conditioned on defect trajectory estimates, which may outperform MWPM-based APM at high defect densities.

7. Conclusion

We have presented the first systematic study of mobile topological defects in surface codes. Our key findings are: (1) defect velocity follows an anomalous power law $v \sim \gamma^{1.7}$ due to syndrome-density feedback; (2) the fault-tolerance phase boundary contracts non-linearly in the (p, ρ) plane, with an analytic lower bound derived from percolation theory; (3) mobile defects induce braiding anomalies that amplify logical error rates by up to 2.4×; and (4) Adaptive Puncture Management with Kalman-filter trajectory prediction recovers 89% of the clean-code threshold under continuous qubit loss. These results provide both fundamental insight into the topological physics of defect-punctured codes and practical guidance for the design of fault-tolerant quantum processors operating under realistic imperfection landscapes.

Acknowledgements

The authors thank the quantum error correction community for stimulating discussions at QIP 2026. Simulations were carried out on the university high-performance computing cluster. A.R.H. acknowledges support from the NSF Graduate Research Fellowship. P.N.S. acknowledges support from the DARPA Quantum Benchmarking programme. K.M.A. and L.T.N. acknowledge support from DOE Office of Science, Basic Energy Sciences.

References

- [1] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.* **303**, 2–30 (2003).
- [2] E. Dennis, A. Kitaev, A. Landahl, J. Preskill, "Topological quantum memory," *J. Math. Phys.* **43**, 4452–4505 (2002).
- [3] A. G. Fowler, A. C. Whiteside, L. C. L. Hollenberg, "Towards practical classical processing for the surface code," *Phys. Rev. Lett.* **108**, 180501 (2012).
- [4] Z. Chen et al., "Exponential suppression of bit or phase errors with cyclic error correction," *Nature* **595**, 383–387 (2021).
- [5] C. Ryan-Anderson et al., "Implementing fault-tolerant entangling gates on the five-qubit code and the color code," arXiv:2208.01863 (2022).
- [6] F. Arute et al., "Quantum supremacy using a programmable superconducting processor," *Nature* **574**, 505–510 (2019).
- [7] T. M. Stace, S. D. Barrett, A. C. Doherty, "Thresholds for topological codes in the presence of loss," *Phys. Rev. Lett.* **102**, 200501 (2009).
- [8] T. M. Stace, S. D. Barrett, "Error correction and degeneracy in surface codes suffering loss," *Phys. Rev. A* **81**, 022317 (2010).
- [9] J. M. Auger, H. Anwar, M. Gimeno-Segovia, T. Rudolph, D. E. Browne, "Fault-tolerance thresholds for the surface code with fabrication errors," *Phys. Rev. A* **96**, 042316 (2017).
- [10] H. Bombin, "Topological order with a twist: Ising anyons from an Abelian model," *Phys. Rev. Lett.* **105**, 030403 (2010).
- [11] B. J. Brown, K. Laubscher, M. S. Kesselring, J. R. Wootton, "Poking holes and cutting corners to achieve Clifford gates with the surface code," *Phys. Rev. X* **7**, 021029 (2017).
- [12] E. Dennis, A. Kitaev, A. Landahl, J. Preskill, "Topological quantum memory," *J. Math. Phys.* **43**, 4452 (2002). [Random-bond Ising mapping, *ibid.*]
- [13] D. S. Wang, A. G. Fowler, A. C. Whiteside, L. C. L. Hollenberg, "Graphical algorithms and threshold error rates for the surface code," *Quantum Inf. Comput.* **10**, 456–469 (2010).
- [14] A. G. Fowler, "Optimal complexity correction of correlated errors in the surface code," arXiv:1310.0863 (2013).
- [15] D. Stauffer, A. Aharony, *Introduction to Percolation Theory*, 2nd ed. (Taylor & Francis, 1994).
- [16] Google Quantum AI, "Suppressing quantum errors by scaling a surface code logical qubit," *Nature* **614**, 676–681 (2023).

- [17] M. Pretko, L. Radzihovsky, "Fracton topological order, generalized lattice gauge theory, and duality," *Phys. Rev. B* **98**, 115134 (2018).
- [18] J. Lisenfeld et al., "Electric field spectroscopy of material defects in transmon qubits," *npj Quantum Inf.* **5**, 105 (2019).
-

Appendix A: Simulation Parameters

All simulations use the following default parameters unless otherwise stated in the main text. Code distances: $d \in \{3, 5, 7, 9\}$. Physical error rate sweep: $p \in [0.01, 0.18]$, 22 equally-spaced points. Defect density sweep: $\rho \in [0.0, 0.25]$, 20 equally-spaced points. Trials per point: 250 (threshold), 200 (defect density), 80 (phase diagram), 30 (velocity). Syndrome coupling: $\alpha = 3$. Velocity measurement: 40 time steps, 20 trials per hop rate, 12 hop rate values in $[0.01, 0.20]$. All random number generation uses NumPy with seed 42 for reproducibility.

Appendix B: Derivation of the Velocity Law

Let $n(t)$ be the number of triggered syndrome bits at time t . The effective hop rate is $\gamma_{\text{eff}} = \gamma(1 + \alpha n/|S|)$. For a single defect in a code at error rate p , the expected syndrome density near the defect is $s \approx 2p(1-p) + p_{\text{defect}}$, where $p_{\text{defect}} \approx \gamma/4$ accounts for boundary-stabiliser triggering by the moving defect itself. Substituting:

$$\gamma_{\text{eff}} \approx \gamma + \alpha \cdot \gamma^2/4 + O(\gamma p)$$

The MSD rate (velocity) is $v = 4 \cdot (\gamma_{\text{eff}}/4) \cdot a^2 = \gamma_{\text{eff}} \cdot a^2$, where $a = 1$ is the lattice spacing. Hence $v \approx \gamma + (\alpha/4)\gamma^2$. In the intermediate regime $\gamma \ll 1$ but $(\alpha/4)\gamma$ dominant, this gives $v \sim \gamma^2$. The observed exponent $\beta \approx 1.7$ falls between 1 and 2, consistent with the crossover between diffusion- and feedback-dominated regimes at finite γ .