Sizing and Capacity Boundaries of Spatial Convolutional Decoders for Surface-Code Quantum Error Correction

Brian Guarraci
Galatheus
brian@galatheus.com

May 2026

Abstract

We study direct neural decoding for the rotated surface code by embedding detector events as a three-dimensional volume: two spatial dimensions from the code lattice and one temporal dimension from syndrome measurement rounds. A residual 3D convolutional network operates on this volume and is conditioned on the physical error rate, so one model covers a range of sub-threshold operating points. The purpose is not to claim state-of-the-art MWPM beating—hybrid and graph-based neural decoders already establish that—but to characterise, with full multi-seed coverage, *how far a pure spatial decoder scales and where it stops*. Across three seeds and a disjoint test split, the decoder beats minimum-weight perfect matching (MWPM) at distances $d = 3, 5, 7$ (relative improvement +11 to +26% at physical error rate $p = 0.003$), reaches a noisy boundary at $d = 9$ at the selected knee (tied with MWPM, large seed-to-seed variance)—which a 5×-data follow-up resolves, beating MWPM by +36 to +57% (three seeds) and showing the $d = 9$ boundary is under-resourcing rather than a fundamental limit—and is under-capacity at $d = 11$ (well below MWPM at the selected knee). We size the network per distance by an empirical width × depth knee, and compare the classic ResNet block against a Leela-Chess-Zero-style block ("LC0"): block design is *not* second-order but distance-dependent—LC0 is neutral at $d = 3, 5$, a clear win at $d = 7$ (+16% over the classic block at $p = 0.003$), and no help at the $d = 9/11$ boundary. We position the work explicitly as a sizing-and-boundary study behind existing LER-beating decoders—not a scaling law and not a new state-of-the-art. All numbers are emitted by a single strict reproduction script over seeds 42/43/44.

1 Introduction

The decoder is the Hamiltonian. A probabilistic prior is literally a Boltzmann distribution: $p(x) \propto \exp(-H(x))$. So any procedure that reconstructs a signal from partial or corrupted observations is an interface to an energy landscape H whose ground state is the correct reconstruction. Classical decoders, neural decoders, generative compression codecs, and autonomous error-correcting hardware differ only in *where* H lives and *how* it is evaluated. Minimum weight perfect matching (MWPM) [7, 8] chooses a fixed combinatorial H read off the code's Tanner graph and finds its ground state by matching. A diffusion model [21] learns the score $\nabla \log p = -\nabla H$ and samples by Langevin descent. A neural QEC decoder amortises $\arg\min_x H(x \mid \text{syndrome})$ into a single forward pass. Autonomous QEC engineers H directly into the physical qubits so that dissipation drives the quantum state toward the codespace ground state, with no classical decoder pipeline at all. All four are interfaces to the same operation: evaluate or descend ∇H of an energy landscape. They differ only in whether H is hand-designed, learned from data, realised as a score field, or baked into hardware.

This framing changes the interesting question. "Can a neural decoder beat MWPM on circuit-level noise?" was the question in 2023–2024, and the answer is now yes: AlphaQubit [11] on real Sycamore data, the graph neural network decoder of Lange et al. [16] on circuit-level depolarising noise in simulation, state-space decoders [12], and hybrid classical–neural decoders such as NMWPM all clear the MWPM bar at the distances they were trained for. What these works do not address is the more fundamental question: **how much capacity does the learned H need as a function of code distance, and does that requirement depend on the architecture of the network that represents H?** Prior work sizes models ad hoc – Lange et al. use a 1.36M-parameter network for $d \leq 7$ and a 2.35M-parameter network for $d = 9$ [16]; NMWPM uses a constant 3.9M parameters across all distances. There is no principled prescription.

Contribution. We give an empirical, multi-seed characterisation of how a pure 3D convolutional surface-code decoder must be *sized* as a function of code distance, and where the pure-spatial approach runs out. Concretely: (i) a per-distance capacity knee in width×depth, selected on validation by a predeclared rule and reported with three-seed (42/43/44) test-set error rates and 95% confidence intervals; (ii) the resulting operating map—MWPM-beating at $d = 3, 5, 7$, a high-variance boundary at $d = 9$, and an under-capacity regime at $d = 11$; and (iii) an architecture comparison (the classic ResNet block [19] versus an LC0-style pre-activation, dual-output squeeze-excite, FiLM-conditioned, ReZero block) showing that block design is distance-dependent, not second-order. We deliberately do *not* fit a closed-form scaling law. The detector volume grows as d^3, which motivates why capacity must grow with distance, but the multi-seed data show a capacity *boundary*—a regime where seed variance explodes and the pure-spatial decoder falls below MWPM—rather than a clean power law, and we report it as such.

Positioning against prior art. "Neural decoder beats MWPM" is not the contribution of this paper. Lange et al. [16] beat MWPM on the same rotated surface code under circuit-level depolarising noise at the same sub-threshold operating points we use, with larger margins than the pure-CNN decoder presented here achieves at matched distances. AlphaQubit [11] beats MWPM on real hardware data, and NMWPM [9] beats MWPM near threshold. Closest to our architecture, Gicev et al. [17] study fully-convolutional 3D networks on the same spatiotemporal detector volume (including a generative variant) and generalise a fixed-size, translation-invariant model across distance; they do not study a capacity-versus-distance sizing rule, which is exactly the gap we fill. What the present work adds is therefore orthogonal to the logical-error-rate race: (i) an empirical, multi-seed capacity-sizing map for a pure spatial decoder as a function of code distance; (ii) an explicit characterisation of the distance at which the pure-spatial approach stops being competitive; and (iii) an architecture comparison testing whether block design matters once capacity is sized—it does, and in a distance-dependent way.

This paper is one member of a four-paper program on learned recovery over constraint systems. The companion occupancy-grid compression work [27] isolates the constraint-locality principle: the parity-check structure must be engineered so that local learned proposals can use syndrome information without being damaged by high-weight algebraic projection. The gauge-structure note shows the complementary warning that an exact algebraic characterisation of a codespace need not prescribe a useful architecture. Surface-code QEC provides the positive-control case for locality: the stabilizer geometry supplies local, overlapping checks by construction. The present paper therefore asks a different question: once locality is guaranteed by physics, how much neural capacity is needed to exploit it as code distance grows?

What this paper is not. We are explicit about the scope so that reviewers can calibrate the contribution:

- **Not autonomous QEC.** All H in this paper is learned by a classical network from simulated data; none is engineered into quantum hardware. The Hamiltonian framing points at autonomous QEC as the natural endpoint of the programme, but the present paper stays in the learned-decoder regime.
- **Not a closed-form scaling law.** We report an empirical per-distance capacity knee and a boundary, not a fitted $\alpha \cdot d^3$ law. The d^3 detector volume motivates why capacity must grow with distance, but the data show a boundary, not a power law; a first-principles capacity bound is a separate problem.
- **Not real hardware.** All syndromes are generated by Stim [18] under circuit-level depolarising noise. We do not test on biased noise, leakage, crosstalk, or real-device syndromes.
- **Not real-time.** Inference is measured on commodity GPUs and remains roughly two orders of magnitude too slow for the syndrome cycle time of current superconducting hardware. The $O(d^3)$ scaling and regularity of the 3D convolutional kernel make the architecture FPGA/ASIC friendly, but we have not built that deployment.

Roadmap. Section 2 summarises the rotated surface code, detector events, and the MWPM baseline. Section 3 describes the 3D convolutional decoder and the error-rate conditioning that lets one learned H serve a continuous range of operating points. Section 4 presents the per-distance capacity knees, the three-seed test-set results with confidence intervals, the resulting MWPM-beating / boundary / under-capacity operating map, and the classic-versus-LC0 comparison. Section 5 discusses follow-ups the results motivate: physics-informed priors that bake code geometry into the decoder before any data is seen, and the LC0 component ablation and paired LC0-versus-classic test that the distance-dependent block-design finding calls for.

2 Background

2.1 Surface code basics

The rotated surface code [2, 4] at distance d encodes one logical qubit in d^2 data qubits with $d^2 - 1$ stabilizers arranged in a checkerboard on a 2D grid. Stabilizers are weight-4 in the bulk and weight-2 at the boundary. X errors flip Z stabilizers and vice versa, so X and Z decoding decouple under simple noise models. The code has distance d, meaning any logical error requires at least d physical errors, and its error suppression improves exponentially with d below the code's threshold error rate [3, 4].

2.2 Detectors and detector events

In modern descriptions following Gidney [18], syndrome processing is described in terms of *detectors*—parities of pairs of consecutive stabilizer measurements that should be equal in the absence of errors. A flipped detector indicates that an error occurred between the two measurements. For distance d with d rounds of measurement, the number of detectors is approximately $d \times (d^2 - 1)$. For each shot, an "observable flip" indicates whether the encoded logical qubit has been flipped relative to its initial state.

2.3 MWPM baseline

We use **PyMatching v2** [7, 8], the standard fast MWPM implementation, configured from Stim's detector error model with `decompose_errors=True`. This is the standard QEC benchmark baseline used across the community [5, 6, 11, 12].

3 Method

3.1 Spatial embedding of detector events

Stim [18] assigns each detector a coordinate tuple (x, y, t) where x, y are the spatial position of the stabilizer on the code lattice and t is the measurement round. We extract these coordinates from the circuit and discretize each axis to integer indices, yielding a grid size (D_x, D_y, D_t). For the rotated surface code with d rounds at distance d, this gives a $(d{+}1) \times (d{+}1) \times (d{+}1)$ grid (sparse, checkerboard occupancy).

For each shot, we construct a sparse binary volume $V \in \{0, 1\}^{D_x \times D_y \times D_t}$ where $V[i, j, k] = 1$ if and only if the detector at coordinate (i, j, k) fired. Cells without an associated detector are zero.

3.2 Architecture

The decoder is a 3D residual convolutional network based on the ResNet architecture [19] adapted to volumetric data:

- **Input convolution** $(1 \to W)$ lifts the sparse binary volume to a feature volume of constant width W
- N **residual blocks**, each containing two `Conv3d` layers, group normalization [20], SiLU activation, and a residual skip connection
- **Error rate conditioning**: a small MLP maps the scalar physical error rate p to a conditioning vector that is added to the per-block feature representations, analogous to the timestep embedding used in diffusion models [21]
- **Global average pooling** over spatial dimensions
- **Output MLP head** producing a single logit for logical flip prediction

The historical fixed-size prototype used $W = 128$ and $N = 12$ (10.7M parameters). The active paper uses per-distance knee cells selected by the clean capacity scan, because fixed capacity is exactly the failure mode being studied.

3.3 Mixed error rate training

A novel feature of our training is that **a single model is trained on a uniform mixture of physical error rates**. For each training iteration, samples are drawn from circuits with $p \in \{0.001, 0.002, \ldots, 0.007\}$, and each sample is paired with its true p as conditioning input. The model learns to adapt its decoding behavior to the operating point. This is the same approach we use for variable-rate compression in our companion paper [27].

3.4 Coset equivalence

Two error patterns that differ by a stabilizer element produce the same syndrome and the same logical effect [3, 4]. The decoder need only identify the coset of the error in the quotient by the stabilizer group. Because our decoder predicts the logical observable flip directly (a single binary outcome) rather than the full error pattern, stabilizer-equivalent errors automatically receive identical training labels. The decoder is therefore invariant to coset choice by construction, without requiring explicit augmentation. This is the same property that other binary-classification QEC decoders rely on, including AlphaQubit [11].

3.5 Training details

We train with AdamW [22], learning rate 10^{-3}, cosine annealing [23] over 30 epochs. Loss is binary cross-entropy between predicted and true observable flip. Training data consists of 200K

samples per error rate × 7 rates = 1.4M total, generated from Stim [18] circuit-level depolarizing circuits. Batch size is 2,048. Multi-seed experiments use seeds 42, 43, 44.

4 Experiments

4.1 Setup

We evaluate on the rotated surface code at distances $d \in \{3, 5, 7, 9, 11\}$ with circuit-level depolarizing noise. Training and validation circuits are generated using Stim [18] with all four common noise sources: data depolarization before each round, Clifford depolarization after each gate, reset bit flip errors, and measurement bit flip errors, all at the same physical error rate p. MWPM baselines are computed using PyMatching v2 [7, 8] with `decompose_errors=True` on the same detector events, enabling paired significance testing via McNemar's test.

The active clean experiment uses the smallest per-distance knee cell that is within the predeclared tolerance of the best seed-42 capacity-scan validation result. Each reported cell must have seeds 42, 43, and 44. Validation uses **1M shots per error rate** per seed, sampled independently from training data.

Data provenance: all results are from k8s Jobs on the galatheus cluster. The runs used RTX 3090, RTX 4070 Ti, and RTX 4090 GPUs. Job logs are archived on the shared scratch PVC under `/work/qec_runs/`.

4.2 Per-distance capacity knees and multi-seed results

For each distance we select the smallest width × depth cell whose seed-42 validation mean logical error rate (LER) is within a predeclared tolerance of the best cell in the capacity scan (the "knee"). The selection rule—minimum mean validation LER across the four eval rates, equally weighted—is fixed before any test-set evaluation. The selected knees are listed in Table 1. Every reported cell is then evaluated on a *disjoint* test split (sampler seed seed × 1000 + 800, disjoint from train and validation) at all three seeds 42/43/44. All numbers below are emitted by `reproduce_qec_results.py` from the per-run `eval_test_split.py` JSONs; the reproducer fails on incomplete seed coverage by construction.

Table 1: Selected per-distance capacity knees (width × depth), each evaluated at seeds 42/43/44 on the disjoint test split.

Block	d=3	d=5	d=7	d=9	d=11
Classic	16×4	64×8	64×8	96×12	96×12
LC0	16×4	64×8	96×12	128×12	96×12

Operating map: win, boundary, failure. Table 2 reports the headline at $p = 0.003$: the three-seed mean test LER for each block against the paired MWPM baseline on the same shots. The pure spatial decoder beats MWPM at $d = 3, 5, 7$, is statistically tied at $d = 9$ (the seed-level 95% CI on the classic LER, $[0.00045, 0.00072]$, straddles the MWPM rate 0.000557), and falls well below MWPM at $d = 11$ at the selected knee. The $d = 9$ and $d = 11$ rows are dominated by seed variance: the per-seed classic $d = 9$ LERs are $\{0.00059, 0.00045, 0.00072\}$ (one seed beats MWPM, two do not), and the LC0 $d = 9$ LERs are $\{0.00060, 0.00049, 0.00139\}$ with a 2–3× outlier. This is the signature of a capacity boundary, not a smooth power-law trend—which is why we report a boundary rather than fit a law.

The $d = 9$ boundary is under-resourcing, not a fundamental limit. That boundary is a property of the *selected knee* (the predeclared width/depth), not of the distance. A follow-up capacity probe at $d = 9$—same architecture (classic, w=96, depth 12) but 5× the training data—*beats* MWPM across the board on the disjoint test split over three seeds $\{42, 43, 44\}$: three-seed mean neural LER 0.000232 vs. MWPM 0.000537 at $p = 0.003$ (+56.5%; per-seed McNemar $p \sim 10^{-7}$–10^{-11}), 0.00346 vs. 0.006785 at $p = 0.005$ (+49%), and 0.0201 vs. 0.0312 at $p = 0.007$ (+36%), with tight cross-seed agreement. So the $d = 9$ "boundary" of Table 2 is an under-training/under-data artifact of the knee, not a fundamental capacity wall—a sizing problem more data resolves. (Numbers emitted by `reproduce_qec_results.py` on the `qec-bnd-may30` probe; `data/d9_dataarm_reproduction.json`, see `REPRODUCIBILITY.md`.)

Table 2: Operating map at $p = 0.003$: three-seed mean test-set LER vs. the paired MWPM baseline (same shots), with relative improvement (positive means the neural decoder beats MWPM). "Improv." is the seed-mean of the per-seed relative improvement, which at the high-variance $d = 9, 11$ boundary differs from the ratio of the mean LERs. Per-seed McNemar tests favour the neural decoder at $d \leq 7$ and MWPM at $d = 11$ (per-seed p-values in the reproduction JSON).

		Classic		LC0	
d	MWPM LER	Neural LER	Improv.	Neural LER	Improv.
3	0.006587	0.005681	+13.7%	0.005657	+14.1%
5	0.003290	0.002734	+16.9%	0.002667	+18.9%
7	0.001396	0.001237	+11.4%	0.001037	+25.6%
9	0.000557	0.000588	−5.6%	0.000827	−49.3%
11	0.000203	0.001049	−453%	0.002455	−1192%

Block design is distance-dependent, not second-order. The two blocks are indistinguishable at $d = 3, 5$ (LC0 is +0.4% and +2.4% over classic at $p = 0.003$), but at $d = 7$ the LC0 block is a clear win: +16.1% over the classic block, i.e. +25.6% vs. MWPM against the classic block's +11.4%. At the $d = 9/11$ boundary LC0 does not help and is in fact worse than the classic block. The architectural choice is therefore not second-order: it extends the useful regime at intermediate distance and offers no rescue at the boundary. This is an aggregate (unpaired) comparison—the two blocks were trained on different shots; a paired same-shot McNemar test between blocks (`eval_paired_comparison.py`) is future work.

Full per-rate results. Table 3 gives the three-seed mean test LER (±SEM) for both blocks across all four physical error rates. The qualitative picture is rate-stable—the win at $d \leq 7$, the $d = 9$ boundary, and the $d = 11$ collapse hold across $p \in \{0.001, 0.003, 0.005, 0.007\}$—and the SEM at $d = 9, 11$ is an order of magnitude larger than at $d \leq 7$, quantifying the boundary's instability.

4.3 Inference latency and scaling

For practical QEC deployment, decoder latency is as important as accuracy: superconducting hardware produces syndromes at microsecond rates, and the decoder must keep up with the cycle time to enable real-time fault-tolerant computation.

A key advantage of our spatially-embedded convolutional approach over attention-based decoders is its scaling behavior with code distance (Table 4).

Our cost matches MWPM and Mamba and is **strictly better than AlphaQubit by a factor of** d. At $d = 11$ this is already an order-of-magnitude difference; at $d = 21$—the distance currently targeted by industry roadmaps—the gap is approximately 21×.

Table 3: Three-seed mean test-set LER (±SEM) at the selected knees (seeds 42/43/44; disjoint test split).

Block	d	p=0.001	p=0.003	p=0.005	p=0.007
Classic	3	0.00064±0.00001	0.00568±0.00002	0.01543±0.00005	0.02926±0.00006
Classic	5	0.00010±0.00000	0.00273±0.00003	0.01276±0.00021	0.03413±0.00060
Classic	7	0.00001±0.00000	0.00124±0.00005	0.01002±0.00041	0.03615±0.00100
Classic	9	0.00000±0.00000	0.00059±0.00008	0.00769±0.00041	0.03793±0.00131
Classic	11	0.00001±0.00001	0.00105±0.00042	0.01235±0.00335	0.06007±0.01058
LC0	3	0.00065±0.00001	0.00566±0.00003	0.01541±0.00004	0.02912±0.00022
LC0	5	0.00010±0.00000	0.00267±0.00004	0.01276±0.00015	0.03398±0.00027
LC0	7	0.00001±0.00000	0.00104±0.00002	0.00881±0.00016	0.03287±0.00057
LC0	9	0.00000±0.00000	0.00083±0.00028	0.00964±0.00242	0.04406±0.00775
LC0	11	0.00003±0.00001	0.00246±0.00077	0.02314±0.00512	0.09092±0.01301

Table 4: Asymptotic per-shot inference cost vs. code distance.

Decoder	Architecture	Per-shot cost	Notes
MWPM	matching	$\sim O(d^3)$	average case
AlphaQubit	recurrent transformer	$O(d^4)$	attention quadratic in d^2
Mamba	state-space	$O(d^3)$	linear per round
Ours	**3D ResNet**	$\boldsymbol{O(d^3)}$	**3D conv over** $(d{+}1)^3$

We measured inference cost on an NVIDIA RTX 4070 Ti (Table 5).

Table 5: Measured inference latency (width 128, depth 12).

d	Grid	Detectors	Batched (μs/shot)	Shots/sec	Single (μs)
3	4^3	24	49	20,243	2,371
5	6^3	120	181	5,528	2,462
7	8^3	336	429	2,333	2,324
9	10^3	720	835	1,198	2,344
11	12^3	1,320	1,441	694	3,001

The batched latency scales as $\sim d^{2.9}$, slightly better than the theoretical d^3 because small distances do not saturate GPU compute. Single-shot latency is approximately constant at 2.3–3.0 ms, dominated by GPU kernel launch overhead.

Path to real-time deployment. Our prototype is approximately 180× slower than the QEC cycle time at $d = 5$. The convolutional architecture is well-suited to standard acceleration: TensorRT compilation (2–5×), model distillation (10–100× at small d), INT8 quantization (2–4×), and FPGA/ASIC synthesis. A conservative 50× improvement would reach $\sim$10 μs at $d = 5$.

4.4 Comparison with prior work

We do not claim state-of-the-art logical error rates, and the comparison to prior neural decoders is deliberately at the level of regime rather than a single headline LER. AlphaQubit [11], Lange et al. [16], Mamba-style state-space decoders [12], and NMWPM [9] already beat MWPM in their respective settings, and Lange et al. do so by larger relative margins than the pure-CNN decoder here achieves at matched distance. Our contribution is orthogonal: the per-distance sizing map, the boundary location, and the distance-dependent block-design effect. The fully-convolutional 3D decoders of Gicev et al. [17] are the closest architecture; they rely on a fixed-size, translation-

invariant network to generalise across distance, whereas we quantify the capacity a non-shared per-distance model needs and the distance at which even that runs out.

4.5 Limitations

We are explicit about what we have and have not shown:

- All experiments are on **simulated** circuit-level depolarizing noise. We have not tested on real hardware data, biased noise, leakage, or crosstalk.
- The $d = 9$ and $d = 11$ results have large seed-to-seed variance (SEM an order of magnitude larger than at $d \leq 7$); the boundary is reported as such, and the $d = 11$ knee is explicitly under-capacity rather than a tuned best case.
- The classic-versus-LC0 comparison is aggregate (unpaired): the two blocks were trained on different shots, so the block-design effect is reported at the level of mean LER, not a paired same-shot test.
- The LC0 win at $d = 7$ is not decomposed into its components (pre-activation, dual-output squeeze-excite, FiLM, ReZero); a component ablation is future work.
- AlphaQubit's headline 25% result is on real hardware (Sycamore data not publicly available); our simulation result is not directly comparable.

5 Connection to Constraint Locality

Our companion compression paper isolates the constraint-locality principle in a setting where the parity checks are a design choice rather than a physical given [27]. In that domain, fixed high-weight projection can damage a strong learned proposal, while local overlapping checks can provide useful refinement signals.

Surface-code decoding gives the corresponding positive-control case. Stabilizers have exactly the required structure *by construction*: each stabilizer acts on at most four adjacent qubits (weight-4 in the bulk, weight-2 at the boundary), neighboring stabilizers share support on data qubits, and a single-qubit error flips at most two neighboring detectors. The compression problem must engineer locality; the surface code supplies it through its geometry.

This connection motivates the present capacity-scaling question. If local overlapping constraints are the regime where learned syndrome decoders can use spatial structure, then a surface-code decoder should expose the detector geometry to the network rather than flattening the syndrome. Our 3D convolutional embedding does exactly that. The remaining question is not whether locality is present, but how much neural capacity is needed to exploit it as the detector volume grows with distance.

6 Discussion and Future Work

What the boundary means, and what it does not. It is tempting to read the per-distance knees as a closed-form law $w \cdot \text{depth} \approx \alpha \cdot d^3$, since the detector volume the network must process grows as d^3. We resist this reading. With three seeds at every distance, the data show a *boundary*, not a power law: the knee is well defined and MWPM-beating through $d = 7$, but at $d = 9$ the decoder is statistically tied with MWPM with seed variance an order of magnitude larger than at smaller distances, and at $d = 11$ the same width $\times$ depth budget that nearly suffices at $d = 9$ is far under capacity. A clean α would require the knee to track d^3 *through* the boundary; instead the boundary is precisely where a fixed sizing heuristic breaks down. We therefore report an empirical operating map and a boundary location, and leave a first-principles capacity bound—ideally from information-theoretic limits given syndrome sparsity and the noise rate—as an open problem.

Block design is distance-dependent. We had hypothesised that, once capacity is sized, the residual block design would be second-order—that the capacity requirement is a property of the decoding problem rather than of the network. The multi-seed data refute the strong version of this. The LC0 block is neutral at $d = 3, 5$, a clear win at $d = 7$ (+16% over the classic block), and no help—indeed worse—at the $d = 9/11$ boundary. Block design matters, and matters most at intermediate distance. Which LC0 component carries the $d = 7$ win (pre-activation, dual-output squeeze-excite, FiLM conditioning, or ReZero) is the most immediate follow-up; it requires block variants that isolate each component and a paired same-shot test against the classic block.

$d = 11$ is under capacity, not an achieved match. At the $d = 11$ knee (w=96, depth=12) the three-seed classic test LER at $p = 0.003$ is 0.00105 versus MWPM's 0.00020—roughly 5× worse, and worse still for LC0. We make no MWPM-match claim at $d = 11$ with the capacities run; the $d = 11$ rows are under-capacity data points that mark where the pure-spatial approach stops and where a larger-capacity or hybrid decoder would be required.

The most promising follow-up: a physics-informed decoder. The boundary suggests the pure-spatial decoder spends much of its capacity re-learning structure the code geometry already specifies. A physics-informed decoder parameterises $H(x) = H_{\text{code}}(x) + H_\theta(x)$, with H_{code} fixed from the code (a differentiable form of the MWPM edge weights) and only the residual H_θ learned. This is the direction in which hybrid decoders already beat MWPM more decisively than pure-neural ones: Lange et al. [16] and especially NMWPM [9], which learns MWPM edge weights and reports near distance-independent parameter efficiency. Because NMWPM already realises much of this idea, the novelty of a physics-informed variant must be argued against it directly rather than against MWPM. The sizing map here predicts where the boundary lies for the pure-spatial decoder, and thus where such a prior would buy the most; collapsing the $d \geq 9$ boundary is the concrete target.

Generative and score-matching decoders. A score- or diffusion-style head that emits a per-cell gradient rather than a single logit is a natural denser-signal alternative, with a test-time compute knob via iterative refinement [21]. This direction is already active: diffusion and masked-diffusion QEC decoders, and the generative variant of the fully-convolutional decoder of Gicev et al. [17], appeared concurrently with this work. We therefore flag it as relevant prior art rather than as our own contribution.

Realistic noise and deployment. Two practical gaps remain that are orthogonal to the Hamiltonian framing. First, all experiments here use circuit-level depolarising noise generated by Stim [18]. Biased noise, leakage, and crosstalk are the regimes in which MWPM's independent-error assumption is most strongly violated; prior neural decoders report their largest improvements there. We expect the same qualitative finding (the learned H captures correlations MWPM cannot) but have not measured it. Second, the prototype is roughly two orders of magnitude too slow for real-time deployment at $d \geq 5$. The $O(d^3)$ cost and the pure-convolutional structure make the architecture suitable for standard accelerator paths (TensorRT, quantisation, FPGA/ASIC synthesis) and we project that a conservative combination reaches the syndrome cycle time at $d \leq 7$. Both gaps are engineering projects rather than research questions; neither bears on the sizing-and-boundary result.

Takeaway. A pure 3D convolutional decoder, sized per distance by an empirical knee, beats MWPM at $d = 3, 5, 7$ on circuit-level depolarising noise, reaches a high-variance boundary at $d = 9$, and is under capacity at $d = 11$. Block design is not second-order: an LC0-style block

extends the win at $d = 7$ but does not rescue the boundary. The useful takeaways are practical—a sizing map for pure-spatial decoders and an honest boundary location—and they point to a hybrid, physics-informed decoder as the way past the boundary, with an LC0 component ablation and a paired block comparison as the immediate next experiments.

7 Conclusion

We presented a multi-seed sizing-and-boundary study of direct 3D convolutional surface-code decoders. Sized per distance by an empirical capacity knee and evaluated on a disjoint test split across seeds 42/43/44, the pure-spatial decoder beats MWPM at $d = 3, 5, 7$, is statistically tied at the high-variance $d = 9$ boundary, and is under capacity at $d = 11$. The residual block design is distance-dependent rather than second-order: an LC0-style block clearly wins at $d = 7$ but does not help at the boundary. We make no scaling-law or state-of-the-art claim; the contributions are an empirical capacity-sizing map for a pure spatial decoder, an honest boundary location, and the block-design finding, all emitted by a single strict reproduction script. The results point to a physics-informed hybrid decoder as the route past the boundary, with an LC0 component ablation and a paired block comparison as the immediate next experiments. Code, trained models, and the reproduction pipeline are released with the paper family in the companion portfolio repository.

References

[1] A. Yu. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.*, vol. 303, no. 1, pp. 2–30, 2003.

[2] S. B. Bravyi and A. Yu. Kitaev, "Quantum codes on a lattice with boundary," arXiv:quant-ph/9811052, 1998.

[3] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *J. Math. Phys.*, vol. 43, no. 9, pp. 4452–4505, 2002.

[4] 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.

[5] R. Acharya et al. (Google Quantum AI), "Suppressing quantum errors by scaling a surface code logical qubit," *Nature*, vol. 614, pp. 676–681, 2023.

[6] R. Acharya et al. (Google Quantum AI), "Quantum error correction below the surface code threshold," *Nature*, vol. 638, pp. 920–926, 2025.

[7] O. Higgott, "PyMatching: A Python package for decoding quantum codes with minimum-weight perfect matching," *ACM Trans. Quantum Computing*, vol. 3, no. 3, pp. 1–16, 2022.

[8] O. Higgott and C. Gidney, "Sparse Blossom: correcting a million errors per core second with minimum-weight matching," *Quantum*, vol. 9, p. 1600, 2025.

[9] N. Peled, S. Zenati, and E. Nachmani, "Neural minimum-weight perfect matching," arXiv:2601.00242, 2026.

[10] N. Delfosse and N. H. Nickerson, "Almost-linear time decoding algorithm for topological codes," *Quantum*, vol. 5, p. 595, 2021.

[11] J. Bausch et al., "Learning high-accuracy error decoding for quantum processors," *Nature*, 2024. (AlphaQubit.)

[12] C. Lee, T. Hur, and D. K. Park, "Scalable neural decoders for practical real-time quantum error correction," arXiv:2510.22724, 2025.

[13] G. Torlai and R. G. Melko, "Neural decoder for topological codes," *Phys. Rev. Lett.*, vol. 119, p. 030501, 2017.

[14] S. Varsamopoulos, B. Criger, and K. Bertels, "Decoding small surface codes with feedforward neural networks," *Quantum Sci. Technol.*, vol. 3, no. 1, p. 015004, 2018.

[15] P. Baireuther et al., "Machine-learning-assisted correction of correlated qubit errors in a topological code," *Quantum*, vol. 2, p. 48, 2018.

[16] M. Lange et al., "Data-driven decoding of quantum error correcting codes using graph neural networks," *Phys. Rev. Research*, vol. 7, p. 023181, 2025.

[17] S. Gicev, L. C. L. Hollenberg, and M. Usman, "Fully convolutional 3D neural network decoders for surface codes with syndrome circuit noise," arXiv:2506.16113, 2025.

[18] C. Gidney, "Stim: a fast stabilizer circuit simulator," *Quantum*, vol. 5, p. 497, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016.

[20] Y. Wu and K. He, "Group normalization," in *ECCV*, pp. 3–19, 2018.

[21] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[22] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[23] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *ICLR*, 2017.

[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, pp. 7132–7141, 2018.

[25] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[26] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Allerton Conf.*, pp. 341–346, 2016.

[27] B. Guarraci, "Constraint locality in learned syndrome decoding for 3D occupancy grids," companion paper, 2026.