SENSITIVITY BEFORE CONNECTIVITY: MINIMAL PAIRS REVEAL HOW CNNS LEARN A GLOBAL VISUAL RULE

Anonymous authors
Paper under double-blind review

ABSTRACT

A network can see every pixel a visual rule depends on and still not use them. We study this gap with seeded connectivity on near-critical 64×64 grids, where a full-access convolutional network faces zero Bayes risk. Path-conditioned AUC suggests that competence spreads from short to long paths during learning, but this picture is misleading. We introduce minimal pairs for spatial rules: two images that differ in a single pixel outside the query's local window, where closing that pixel disconnects the seed from the query. Size-matched placebo edits, which remove at least as much of the query's component without disconnecting it, separate connectivity from cluster size. Minimal pairs reveal three stages. With 64k training examples, the model's discrimination does not exceed what the seed position alone provides, and it ignores cuts. With 256k, it responds to distant cuts at every path length, but mostly in proportion to how many pixels an edit removes. With 1M, it computes connectivity itself, including on paths that need more certified propagation steps than it has layers. Within individual training runs, the response to cuts appears at all path lengths at once. At first, it depends about equally on how many pixels an edit removes and on whether it disconnects. The weight on disconnection then grows until removal size no longer matters. Path length orders the strength of the response rather than its presence, and response strength is what path-conditioned AUC detects.

1 INTRODUCTION

A neural network can receive every pixel that a visual rule depends on and still fail to use them. We study the gap between *seeing* information and *using* it with seeded connectivity: given a binary image, a seed on the border, and a query at the centre, decide whether the two lie in the same four-connected occupied component. Connectedness is a classic global visual predicate (Minsky & Papert, 1969). On near-critical random fields, connecting paths are long and winding, yet once the full image is visible the label is fully determined, so any remaining error is a failure of learning rather than of information.

A natural way to follow how such a rule is acquired is to condition performance on instance difficulty, here the shortest seed-to-query path length ℓ, with a threshold-free metric such as AUC (Schaeffer et al., 2023). Doing so gives a tidy picture (Figure 3). With 256k training examples, a full-access convolutional network separates short and middle paths better than a trained local-window model, but shows no advantage on the longest paths; with 1M examples the advantage covers all paths. Read at face value, nonlocal competence spreads outward along path length.

This picture is misleading. On the longest paths at 256k, path-conditioned AUC shows no advantage over the local model ($\Delta = -0.011$). Yet when a single pixel far outside the local window is closed so that the seed becomes disconnected, the same model lowers its score in 83% of such pairs. The model already responds to distant structure on long paths. AUC misses this because the response is small compared with how much scores vary across unrelated images.

We make this observation systematic with *minimal pairs*. A cut pair consists of a positive image and a copy in which one pixel outside the query's radius-16 window is closed, disconnecting seed and query. Because the two images are identical inside the window, any preference between them requires nonlocal information, and no trained baseline is needed. A size-matched *branch placebo*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

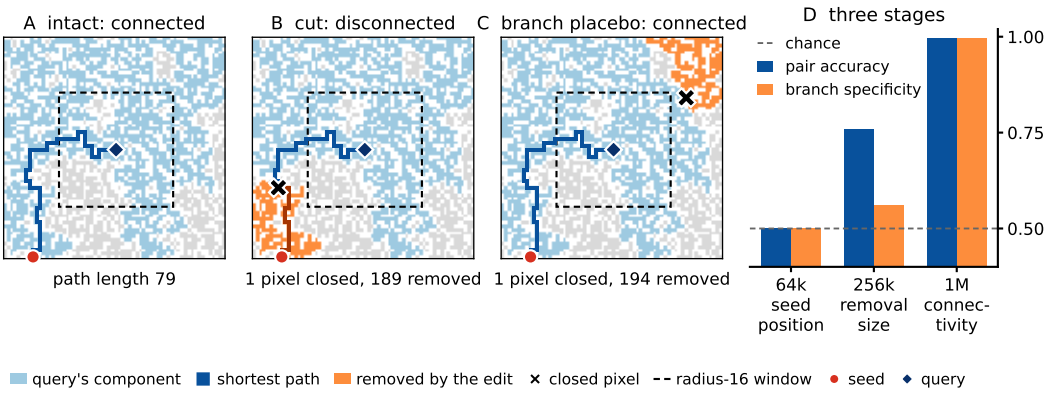


Figure 1: **Minimal pairs for seeded connectivity. A**: a positive image. Light blue marks the query's component, dark blue the shortest seed–query path ($\ell = 79$), and the dashed square the radius-16 window. **B**: cut edit. Closing one pixel outside the window disconnects seed and query; the 189 pixels it separates from the query's component are shown in orange. A model that sees only the window cannot tell A from B. **C**: branch placebo. Closing a different pixel removes slightly more of the component (194 pixels) but leaves seed and query connected, so it matches the cut in size but not in connectivity. **D**: median pair accuracy (the model scores A above B) and branch specificity (it scores C above B) over five initializations. At 64k the model ignores cuts; at 256k it responds to them but barely separates the cut from the placebo; at 1M it separates both. The dashed line is chance.

closes a different pixel that removes at least as much of the query's component while keeping seed and query connected. A score that tracks component size prefers the cut image; a score that tracks connectivity prefers the placebo. An oracle, a window-only scorer, and a component-size proxy all behave exactly as intended on these tests.

Minimal pairs reveal three stages (Table 1). With 64k examples, the model's discrimination does not exceed what the seed position alone provides, and it ignores cuts. With 256k, it responds to distant cuts at every path length, but mostly in proportion to how many pixels an edit removes. With 1M, it computes connectivity: it prefers the still-connected placebo in 99.6% of size-matched pairs, including on paths that need more certified propagation steps than it has layers. Within individual training runs the same progression appears continuously: the response to cuts emerges at all path lengths together, depends at first about equally on removal size and on disconnection, and then increasingly on disconnection alone. Path length does matter, but it orders the *strength* of the response, not its presence, and strength is what path-conditioned AUC detects.

Our contributions are:

1. **Minimal-pair evaluation for spatial rules.** Single-pixel cuts placed where a restricted model provably cannot see them, together with size-matched placebo edits, separate nonlocal use, sensitivity to removal size, and connectivity computation.

2. **A three-stage account of learning**, seen across training scale and within individual runs: seed position, then a size-sensitive nonlocal response, then connectivity-specific computation.

3. **Evidence that path-conditioned AUC misleads about coverage.** Seed position explains the early short-path advantage, sensitivity appears on all path lengths together, and path length orders only the response strength.

4. **Verified connectivity computation beyond a flood-fill budget** in a feedforward CNN, established against size-matched placebos rather than inferred from AUC.

The broader lesson is that difficulty-conditioned metrics measure how far a response stands out from the variability of the data, not which instances a model processes. Minimal pairs measure the latter directly, and they are cheap to build whenever a single edit can flip the label.

2 Related Work

Connectivity and long-range grouping. Pathfinder varies contour length and motivated recurrent and dilated architectures for long-range grouping (Linsley et al., 2018; Kim et al., 2020; Linsley et al., 2020). Larsen & Druckmann (2022) study connectedness as tag propagation and show that shallow feedforward networks can outperform a truncated literal propagation process, while sufficiently deep tied recurrent models become more propagation-like. We ask how competence develops during training once the full image is already available, and we test what the network responds to rather than only how accurate it is.

Algorithmic learning and example difficulty. Neural algorithmic reasoning studies how networks learn graph procedures (Veličković et al., 2020; 2022), and algorithmic alignment links architecture to sample efficiency (Xu et al., 2020). Recurrent models can extrapolate by running additional iterations (Schwarzschild et al., 2021; Bansal et al., 2022), connectivity can be analyzed through the computational depth of Transformers (Sanford et al., 2024), and training data can steer Transformers toward algorithmic rather than heuristic connectivity solutions (Ye et al., 2025). Networks often fit easier examples before harder ones (Arpit et al., 2017; Baldock et al., 2021), and several theories formalize ordered acquisition of target structure (Abbe et al., 2022; 2023; 2024; Michaud et al., 2023). Because thresholded metrics can create misleadingly sharp transitions (Schaeffer et al., 2023), difficulty-conditioned analyses usually rely on threshold-free scores; we show that such scores can still misstate which instances a model processes.

Minimal pairs and counterfactual evaluation. Evaluating models on inputs that differ in a controlled way has a long history in language modeling: contrast sets probe local decision boundaries (Gardner et al., 2020), counterfactually augmented data isolates the features that should change a label (Kaushik et al., 2020), and linguistic minimal pairs test grammatical knowledge (Warstadt et al., 2020). We bring this idea to a global visual rule with two additions: edits are placed where a restricted-access model provably cannot see them, and placebo edits are matched on the most plausible shortcut, component size.

Architecture and training protocol. Dilation and U-Nets alter how information moves across space (Yu & Koltun, 2016; Ronneberger et al., 2015), effective receptive fields can be smaller than structural ones (Luo et al., 2016), and from-scratch training can distort architecture comparisons (Amos et al., 2024). We therefore report realized update counts and treat architecture results as descriptive.

3 Setup

Task and information access. We use 64×64 binary images in which each pixel is occupied independently with probability $q = 0.60$, close to the four-connected site-percolation threshold, where connecting paths become long and tortuous (Zhou et al., 2012). The query is fixed at the centre, $P = (32, 32)$; the seed S is drawn uniformly from the border; both are forced open. The label is $Y = \mathbb{1}\{S \text{ and } P \text{ lie in the same four-connected occupied component}\}$, and for positives ℓ is the number of edges on a shortest connecting path. The frozen test set has $20{,}000$ images with $5{,}256$ positives. Models receive occupancy, a one-hot seed mask, and two constant maps holding the seed coordinates. Let Z_R denote the pixels within Chebyshev radius R of the query together with this context, and $\epsilon^\star_R = \inf_g \Pr[g(Z_R) \neq Y]$ its Bayes risk. At $R = 32$ the whole image is visible, so $\epsilon^\star_{32} = 0$. The seed position alone is informative: across border cells, $P(Y = 1)$ ranges from 0.127 to 0.341, estimated on $500{,}000$ images from unused stream 9901.

Models and training. The main model is a LocalCNN: a 1×1 input projection, L residual units with one 3×3 convolution each, and a centre readout, so its structural radius is $R = L$. We compare $R = 32$ (full access) with $R = 16$ (local window), and also test fixed-parameter dilated CNNs, a tied recurrent CNN, a U-Net, a pixel-token Transformer, and a four-neighbour message-passing network (Sections F and Q). Training pools are nested, $N \in \{10^3, 4{\cdot}10^3, 1.6{\cdot}10^4, 6.4{\cdot}10^4, 2.56{\cdot}10^5, 10^6\}$, with AdamW (learning rate 3×10^{-4}, weight decay 10^{-4}, batch 256, gradient clipping at norm 1), a cap of 80 epochs or 50k updates, and early stopping after $4{,}000$ updates without improvement in

validation BCE. Larger pools therefore also receive more updates, so we treat N as training scale under a common policy (Table 11). Two further $N = 10^6$ LocalCNN runs are trained to 50k updates with checkpoints every 1,000 updates.

Path-conditioned AUC and reference scorers. With $s(x)$ the pre-threshold query logit, positives $\mathcal{P}_b$ in path bin b, and the common set of test negatives $\mathcal{N}$, we compute $\mathrm{AUC}_b = \Pr[s(x^+) > s(x^-)]$ with $x^+ \sim \mathcal{P}_b$, $x^- \sim \mathcal{N}$, and ties counted as one half. The 23 bins are listed in Table 4. We compare against two references: a *seed-only* scorer that assigns each image the estimated $P(Y = 1)$ of its seed's border cell, and a trained $R = 16$ LocalCNN, with $\Delta_b = \mathrm{AUC}_b^{(32)} - \mathrm{AUC}_b^{(16)}$.

Minimal pairs. Pairs are built from 40,000 fresh images from an unused stream (seed 8808). For a positive x^+, a *cut site* is an occupied pixel on a shortest S–P path, outside the radius-16 window, whose closure disconnects S from P. There is no minimum distance from the seed. Among such pixels we take the one nearest the query. Closing it gives x^-_{cut}. We report the *pair accuracy*

$$\mathrm{PA} = \Pr\big[s(x^+) > s(x^-_{\mathrm{cut}})\big] + \tfrac{1}{2}\Pr\big[s(x^+) = s(x^-_{\mathrm{cut}})\big].$$

Because the cut lies outside the radius-16 window, x^+ and x^-_{cut} are identical to any model restricted to that window, which must tie exactly; $\mathrm{PA} > 0.5$ therefore requires information from outside the window. Two placebo edits are scored with the same seed. The *off-path* placebo closes an occupied pixel that lies outside the query's component and outside the radius-16 window, so neither the label nor the size of that component changes. The *branch* placebo closes a pixel of the query's component that is not on the shortest path and not inside the window; the closure must leave S connected to P and must leave the query component no larger than the cut does (Section A). On the conservative subset where the branch edit removes at least as many pixels as the cut, the *branch specificity*

$$B = \Pr\big[s(x^-_{\mathrm{branch}}) > s(x^-_{\mathrm{cut}})\big]$$

asks whether the model prefers the still-connected image. A score that depends only on the size of the query's component yields $B \le 0.5$; a connectivity oracle yields $B = 1$. Finally, the standardized cut effect in bin b is the mean logit drop $s(x^+) - s(x^-_{\mathrm{cut}})$ divided by the standard deviation of logits over test negatives. On validation scorers the tests behave as intended: an oracle gives $\mathrm{PA} = B = 1$, a window-only scorer ties at exactly 0.500 on both, and a component-size proxy has pair accuracy 1.000 but branch specificity 0.011. Every $R = 16$ checkpoint ties at exactly 0.500. In total, 9,555 positives (89.7%) have a qualifying cut site, and 5,863 pairs fall in the conservative branch subset.

Propagation depth. To test whether a network could be propagating a reachability flag one step per layer, we define propagation depth d_{prop} on a graph in which one 3×3 operation certifies either a four-neighbour step or a diagonal step whose shared corner is occupied, so $d_{\mathrm{prop}} \ge \lceil \ell/2 \rceil$. A literal front with L layers can certify only examples with $d_{\mathrm{prop}} \le L$, and fronts grown from both endpoints must meet within $2L$ (Section J).

4 Results

4.1 Seeing is not using

For $R \le 16$, LocalCNNs stay near the constant-negative error at every pool size, consistent with the restricted-observation floor. At $R = 32$ the Bayes error is zero, yet the model stays at the majority solution through 64k (Figure 2). At 256k, all ten initializations recall some positives (0.036–0.206) while natural error remains 0.262–0.270. At 1M, four of five runs reach errors of 0.017–0.040 with recall 0.89–0.95, and the fifth reaches 0.112 with recall 0.72 (Table 16). Pool size and optimization budget are entangled: selected checkpoints come from about 1.3k updates at 64k, 8.8–10k at 256k, and 43–48k at 1M (Table 11). Cross-pool comparisons below are therefore descriptive; the within-run analyses in Section 4.4 hold the pool fixed.

4.2 What path-conditioned AUC shows, and what it hides

At 64k, full-access AUC falls from 0.776 in the shortest bin to 0.490 in the longest, and the $R = 16$ model follows almost the same curve (0.752 to 0.484). Seed position explains both. A scorer that

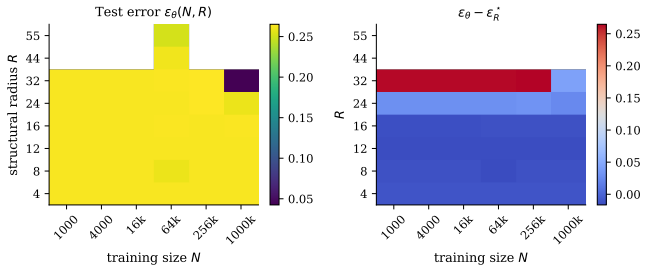


Figure 2: **Full access arrives before learning.** LocalCNN test error across training-pool size N and structural radius R (three initializations per cell; Table 15). At $R = 32$ the information floor is exactly zero, yet low error appears only at the largest scale.

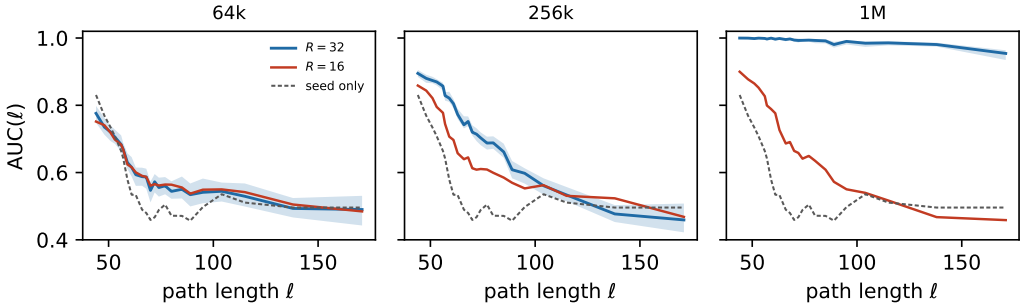


Figure 3: **Path-conditioned AUC against a common negative set.** Blue: full-access LocalCNN ($R = 32$). Orange: trained $R = 16$ local-window model. Grey dashed: seed-position-only scorer. Lines show medians over initializations; bands are 95% image-bootstrap intervals for that median. At 64k both trained models sit at or below the seed-only curve in the shortest bins.

sees nothing but the seed's border cell reaches 0.831 in the shortest bin, above every one of the five 64k $R = 32$ initializations (0.757–0.797). Its advantage fades to 0.53 by $\ell \approx 63$ and dips to about 0.46 near $\ell \approx 70$ and $\ell \approx 89$ (Table 4), which accounts for the below-chance values of the local model in those bins. At 64k, the full-access model has learned no more than where the seed is.

At 256k, the full-access model pulls ahead of the local model on short and middle paths (excess 0.038 in the shortest bin and a median of 0.109 over $\ell \approx 59$–72), but not in the longest bin (-0.011). At 1M the excess reaches 0.496 in the longest bin, where the local model is at 0.459 and the full-access model at 0.954. Within the two 1M runs, the furthest bin with $\mathrm{AUC}_b \geq 0.75$ moves steadily outward during training (Section E).

Read at face value, these curves say that nonlocal competence spreads from short to long paths. But AUC_b compares bin-b positives with the whole negative population, so a response that is present but small relative to the spread of scores across unrelated negatives does not register. Minimal pairs remove that between-image variability.

4.3 Minimal pairs reveal three stages

Table 1 and Figure 4 summarize the pairs. At 64k, pair accuracy and branch specificity are both 0.501: the model ignores distant structure entirely, consistent with its seed-only-level AUC.

At 256k, pair accuracy is 0.759 (five initializations: 0.743–0.816), and the response is not confined to short paths. In the longest bin, pair accuracy is 0.83, in exactly the bin where AUC shows no advantage over the local model. Branch specificity, however, is only 0.560 (initializations 0.546–0.622, each with an image-bootstrap interval of about ± 0.012 that lies above 0.5). When a placebo removes at least as many pixels from the query's component as the cut, the model barely prefers the still-connected image. Its response mostly tracks how much is removed (Section 4.5).

Table 1: **Three stages of learning, measured with minimal pairs.** Medians over five initializations of the full-access LocalCNN. The seed-only scorer reaches 0.831 in the shortest bin. Pair accuracy uses 9,555 cut pairs; branch specificity uses the 5,863 pairs in which the placebo removes at least as many pixels from the query's component as the cut. For reference, a component-size proxy gives branch specificity 0.011 and an oracle gives 1.

Pool	Shortest-bin AUC	Pair accuracy	Branch specificity	What the model uses
64k	0.776	0.501	0.501	Seed position only
256k	0.896	0.759 (longest bin 0.83)	0.560	Distant structure, mostly in proportion to how much is removed
1M	1.000	0.996 (longest bin 0.95)	0.996	Connectivity

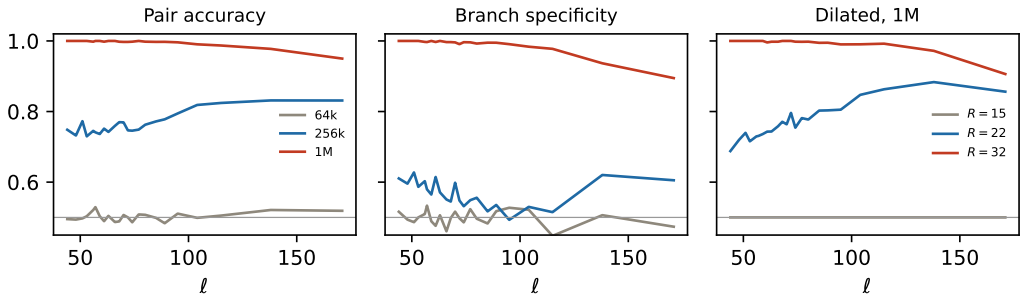


Figure 4: **Minimal pairs by path length.** Pair accuracy (left) and branch specificity (middle) for the full-access LocalCNN at 64k, 256k, and 1M; medians and ranges over initializations. Right: dilated CNNs with equal depth and parameter count at radius 15, 22, and 32, trained on 1M examples. The radius-15 model ties exactly, as it must.

At 1M, pair accuracy and branch specificity are both 0.996, against 0.011 for the component-size proxy. Every path bin is high (longest bin 0.95). Closing an occupied pixel outside the query component and outside the window, which changes neither the label nor the component size, does not systematically lower the score: the five 1M initializations have off-path pair accuracy 0.491–0.510 (median 0.499). At this stage the model computes connectivity rather than cluster size.

Models with restricted radius provide an access check. Every $R = 16$ LocalCNN ties exactly, as it must. A LocalCNN with radius 24 reaches pair accuracy 0.56 at 256k and 0.79 at 1M overall, but that average mixes two populations. Of the 9,555 cuts, 6,689 lie at Chebyshev distance 17–24 from the query and 2,866 lie farther out. On the farther cuts the radius-24 model ties at exactly 0.500 at both pool sizes. On the nearer cuts its median pair accuracy is 0.585 at 256k (three seeds: 0.540, 0.585, 0.625) and 0.916 at 1M (0.888, 0.956, 0.916). Dilated CNNs, which share depth and parameter count across radii, tie at exactly 0.500 with structural radius 15. Radius 22 likewise ties at exactly 0.500 on the 3,726 cuts beyond distance 22, and on the 5,829 cuts inside that radius its median is 0.930 at 256k and 0.944 at 1M. Dilated radius 32 reaches 0.99 at 1M.

4.4 Within a run: from removal size to connectivity

The two checkpointed 1M runs show the same progression without any change of pool size (Figure 5 and Table 6). Up to 3k updates, pair accuracy and branch specificity are at chance in both runs. Pair accuracy leaves chance at 4k (0.518 and 0.524, with bootstrap intervals above 0.5) and then rises quickly: 0.648 and 0.620 at 5k, 0.83 at 8k, and 0.93–0.94 at 15k. At 5k it lies between 0.57 and 0.69 in every one of the 23 path bins, so the response appears at all path lengths together. Branch specificity leaves chance only slightly later (0.524 and 0.519 at 5k) but rises much more slowly: 0.63 at 8k, 0.82–0.83 at 15k, and 0.98 by 30k. Once the two curves separate, their bands do not overlap.

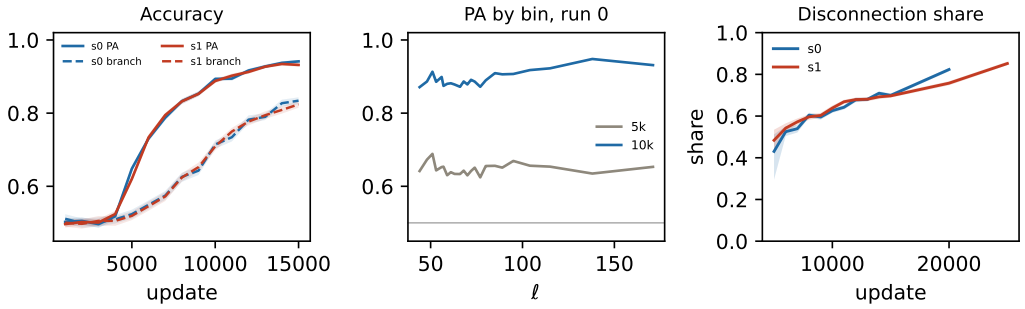


Figure 5: **Two 1M training runs.** Left: pair accuracy and branch specificity over training, with 95% pair-bootstrap bands; checkpoints every 1k updates up to 15k and every 5k afterwards. Middle: pair accuracy by path bin at 5k and 10k updates. Right: share of the predicted logit drop of the median cut (152 pixels removed) that is attributable to disconnection, from a per-checkpoint regression; the share is defined while the size coefficient is positive.

A regression shows what changes, without any threshold. At each checkpoint we regress the logit drop of every edit (cut, branch, and off-path) on $\log(1+\text{pixels removed from the query's component})$ and a disconnection indicator, obtaining β_{size} and β_{conn}. The plotted quantity is the share of the predicted drop of the median cut, which removes 152 pixels, that is attributable to disconnection,

$$\frac{\beta_{\text{conn}}}{\beta_{\text{conn}} + \beta_{\text{size}} \log 153},$$

computed from the unrounded coefficients and defined only while $\beta_{\text{size}} > 0$. Pair-bootstrap intervals are in Table 7 and on the right of Figure 5. The share is unidentified through 3k updates. At 5k it is 0.43 [0.30, 0.51] and 0.48 [0.42, 0.53]. It then rises on both runs: 0.60 [0.59, 0.62] and 0.60 [0.58, 0.61] at 8k, 0.70 [0.69, 0.71] and 0.70 [0.69, 0.71] at 15k, and 0.82 [0.82, 0.83] and 0.76 [0.75, 0.77] at 20k. At 25k in run 0 and 30k in run 1, β_{size} turns negative, so the share is no longer reported: removing more pixels no longer lowers the logit, while β_{conn} keeps growing (to 10.8 at 50k in run 0). The response therefore begins as a mixture of removal size and disconnection and ends as a response to disconnection alone. This also explains why branch specificity starts near 0.5: when a placebo removes at least as many pixels as the cut, the size and disconnection components pull in opposite directions and roughly cancel.

Branch specificity by path bin at 10k favors short paths in one run but not in the other, and the persistent late gap in the final bin rests on 19 pairs; we therefore do not claim a path-length order for connectivity specificity (Section C).

4.5 Path length orders strength, not presence

Why does pair accuracy reach 0.83 on the longest paths at 256k while branch specificity stays near 0.56? Cuts on long paths detach larger pieces of the query's component (median 54 pixels in the shortest bin and 576 in the longest; correlation of ℓ with removal size 0.39), and at 256k the response follows removal size. Pair accuracy is 0.515 when a cut removes 2–48 pixels and rises to 0.722, 0.878, and 0.923 for removals of 48–152, 152–327, and more than 327 pixels (Figure 6a). Holding removal size fixed reverses the path-length pattern. In every stratum above 48 pixels, short paths beat long paths in every initialization, with no overlap between the ranges (0.78–0.90 against 0.64–0.77; 0.92–0.97 against 0.79–0.88; 0.95–0.99 against 0.86–0.92; Table 8). The high long-path pair accuracy at 256k is therefore a removal-size effect.

Response magnitude tells the same story (Figure 6b). At 64k the standardized cut effect is zero (below 1.3×10^{-5} in every bin). At 256k it falls from 0.053 in the shortest bin to 0.0055 in the longest, and at 1M from 2.62 to 0.41, with no overlap between initializations in the extreme bins (Table 9). Because long paths carry larger removals, removal size cannot produce this decline.

These results reconcile the two views. Nonlocal sensitivity appears at all path lengths together (Section 4.4), but it is stronger on short paths. At 256k even the shortest-bin effect is only about

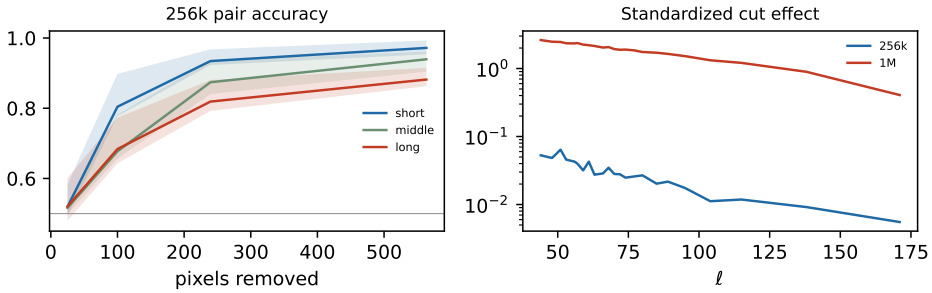


Figure 6: **Removal size and response strength.** (a) Pair accuracy at 256k by the number of pixels the cut removes from the query's component, separately for short, middle, and long paths; medians and ranges over five initializations. (b) Standardized cut effect by path bin at 256k and 1M (log scale); medians and ranges over initializations.

a twentieth of the spread of negative scores, so AUC registers it mainly on short paths, where it is largest and where seed position adds separation of its own. The outward-moving AUC frontier is a frontier of response strength, not of coverage.

4.6 Connectivity beyond a flood-fill budget

A 32-layer LocalCNN that advanced a reachability front by one certified hop per layer could certify only 245 of the 5,256 test positives ($d_{\text{prop}} \leq 32$), and fronts grown from both endpoints would need $d_{\text{prop}} \leq 64$ to meet. Earlier evidence that the network exceeds these budgets came from path-conditioned AUC (Section E), which is open to shortcuts. Minimal pairs close that gap (Figure 7 and Table 3). Across the five 1M initializations, median pair accuracy is 0.999 for $d_{\text{prop}} \in [31, 48)$, 0.997 for $[48, 64)$, 0.991 for $[64, 80)$, 0.982 for $[80, 96)$, and 0.961 for $[96, 128)$, and branch specificity follows it: 0.999, 0.995, 0.980, 0.973, and 0.917. At $d_{\text{prop}} \geq 96$, three times the one-front budget, the network still prefers the still-connected placebo over the cut in most size-matched pairs, although that bin holds only 36 branch pairs (initialization range 0.778–1.000). The network therefore computes connectivity, not cluster size, on paths far beyond what a flood-fill front of its depth could certify, perhaps by encoding larger-scale connectivity summaries across channels. Tied recurrence behaves differently: successful $T = 64$ runs track the number of recurrent steps, consistent with the propagation-like solutions of Larsen & Druckmann (2022) (Section E).

4.7 Scope

The three-stage account is established for the LocalCNN on near-critical Bernoulli fields. Other architectures show different path-conditioned AUC profiles (Section E); tied recurrence reaches low error at 256k but also runs for many more passes over the pool (Table 11). Minimal pairs were not scored for the U-Net or the recurrent models. The pixel-token Transformer stays at the constant-negative solution in every tested setting, which we attribute to the tested from-scratch, early-stopped protocol rather than to attention in general (Section M). The same rule is learned at much smaller pools on smooth shapes and COCO masks (the latter with possible index overlap between training and test masks), while a highly imbalanced room-and-corridor generator stays at the majority solution; a Bernoulli-trained 1M model transfers to COCO masks without error (Table 12).

5 Discussion

Coverage versus strength. Path-conditioned AUC suggested that a model processes short paths before long ones. Minimal pairs show instead that sensitivity to distant structure appears on all path lengths at once and that path length orders its strength. The distinction matters whenever difficulty-conditioned metrics are used to argue about what a model has learned: such metrics measure how far a response stands out from the variability of the data, not which instances the model processes.

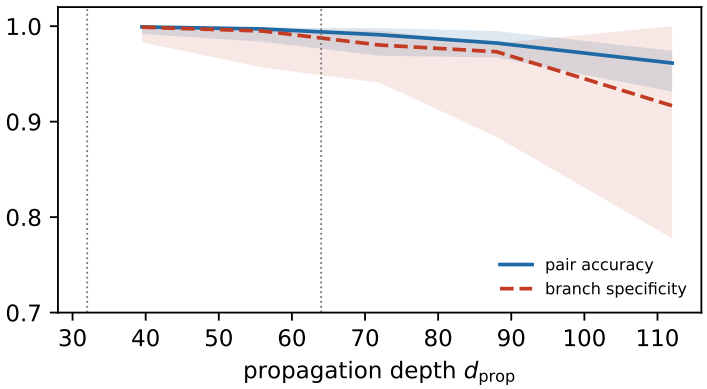


Figure 7: **Minimal pairs by certified propagation depth at 1M.** Pair accuracy and branch specificity against the propagation depth of the positive image; medians and ranges over five initializations. The one-front budget is $d_{\text{prop}} = 32$ and the two-front meeting budget is $d_{\text{prop}} = 64$. Beyond $d_{\text{prop}} = 128$ only four pairs have a branch edit, so the branch curve stops there.

Two kinds of nonlocal use. A model can depend on distant pixels without computing the rule. At 256k, the LocalCNN's response to a cut mostly tracks how much of the query's component disappears. Within a run, the weight on disconnection grows gradually until removal size stops mattering, so the transition is a shift in what the response encodes rather than a switch between two separate mechanisms. Seen through accuracy alone, both stages look like partial connectivity competence.

A reusable tool. Cut pairs need no trained baseline, because the restricted model's response is fixed by construction, and the placebo design makes explicit which shortcut is being ruled out. The recipe applies to any rule where a single edit can flip the label: choose edits outside the region a restricted model can see, and match placebo edits on the most plausible proxy.

Limitations. The within-run results come from two training runs. Under early stopping, pool size and optimization budget move together, so cross-pool comparisons are descriptive; the within-run order does not depend on them. All results use one near-critical distribution and one grid size. Cut pairs exist for 89.7% of positives and involve a single edit, and the branch placebo rules out a component-size proxy but not every conceivable shortcut. Finally, we constrain the mechanism without identifying it.

6 Conclusion

Seeing the whole image is not the same as using it. On near-critical visual connectivity, a full-access CNN first learns only where the seed is. It then responds to distant structure at every path length, at first in proportion to how much an edit removes and increasingly to whether it disconnects, until it computes connectivity itself, well beyond what a flood-fill front of its depth could certify. Path-conditioned AUC turned this sequence into an apparent short-to-long frontier because it measures response strength rather than coverage. Minimal pairs with identical local windows and size-matched placebos separate these possibilities directly.

ARTIFICIAL INTELLIGENCE USE STATEMENT

Generative artificial intelligence tools were used for language editing, manuscript organization, coding assistance, figure preparation, literature search, and consistency checks. All AI-assisted content was reviewed and verified by the authors.

ETHICS STATEMENT

The study uses procedural binary images and COCO 2017 instance masks only as spatial structure, without natural image appearance or class labels. It involves no human subjects and no deployment in decision systems. Section P records the COCO annotation release, mask construction, and the index overlap limitation of the D4 split.

REPRODUCIBILITY STATEMENT

Generators are deterministic functions of a split seed and sample index (Section P). Architectures, side information, and hyperparameters are described in Section 3 and Section Q, parameter counts in Section F, and executed updates in Table 11. Code, including pair construction and the validation scorers, will be released upon acceptance.

REFERENCES

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. The merged-staircase property: A necessary and nearly sufficient condition for SGD learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 4782–4887, 2022.

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. SGD learning on neural networks: Leap complexity and saddle-to-saddle dynamics. In *Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 2552–2623, 2023.

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? the globality barrier and inductive scratchpad. In *Advances in Neural Information Processing Systems*, 2024.

Ido Amos, Jonathan Berant, and Ankit Gupta. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. In *International Conference on Learning Representations*, 2024.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 2017.

Robert J. N. Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems*, 2021.

Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. In *Advances in Neural Information Processing Systems*, 2022.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1307–1323, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117/.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklgs0NFvr.

Junkyung Kim, Drew Linsley, Kalpit Thakkar, and Thomas Serre. Disentangling neural mechanisms for perceptual grouping. In *International Conference on Learning Representations*, 2020.

Brett W. Larsen and Shaul Druckmann. Towards a more general understanding of the algorithmic utility of recurrent connections. *PLOS Computational Biology*, 18(6):e1010227, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014.

Drew Linsley, Junkyung Kim, Vijay Veerabadran, Charles Windolf, and Thomas Serre. Learning long-range spatial dependencies with horizontal gated recurrent units. volume 31, 2018.

Drew Linsley, Alekh Karkada Ashok, Lakshmi Narasimhan Govindarajan, Rex Liu, and Thomas Serre. Stable and expressive recurrent vision models. In *Advances in Neural Information Processing Systems*, 2020.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2016.

Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In *Advances in Neural Information Processing Systems*, 2023.

Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *International Conference on Machine Learning*, 2024.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems*, 2023.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In *Advances in Neural Information Processing Systems*, 2021.

Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. Neural execution of graph algorithms. In *International Conference on Learning Representations*, 2020.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *International Conference on Machine Learning*, 2022.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321. URL https://aclanthology.org/2020.tacl-1.25/.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations*, 2020.

Qilin Ye, Deqing Fu, Robin Jia, and Vatsal Sharan. Transformers provably learn algorithmic solutions for graph connectivity, but only with the right data. *arXiv preprint arXiv:2510.19753*, 2025.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016.

Zongzheng Zhou, Ji Yang, Youjin Deng, and Robert M. Ziff. Shortest-path fractal dimension for percolation in two and three dimensions. *Physical Review E*, 86:061101, 2012.

A Minimal-Pair Construction and Validation

Construction. Images come from stream 8808, which is used for no training, validation, or test split. For each positive we compute one shortest seed–query path. The cut site is the pixel on that path, excluding the endpoints, of smallest Chebyshev distance to the query among those outside the radius-16 window, whose closure disconnects seed from query. There is no further exclusion near the seed. The off-path placebo is an occupied pixel outside the query's component and outside the window, so the edit changes neither the label nor the component size; pair accuracy on this edit is the rate at which the intact image outscores the edit, with ties counted as one half. The branch placebo is chosen from pixels of the query's component that are not on the shortest path, are outside the window, and are neither endpoint. Candidates are examined in random order, up to 16 that lie outside the window. A candidate is eligible only if closing it leaves seed and query connected and leaves the query component no larger than the cut does. Among eligible candidates we keep the one that leaves the smallest query component. Branch specificity uses only pairs in which such a candidate exists (5,863 of 9,555). These rules, and the requirement that a window-only scorer tie at 0.500, were fixed and checked on an oracle, a window-only scorer, and a component-size proxy before any trained checkpoint was scored.

Validation. Table 2 lists the scorers used to check the tests. A model that sees only the radius-16 window must tie exactly, which also checks that inputs are encoded and cropped as in training.

Table 2: Behaviour of reference scorers on the minimal-pair tests.

Scorer	Pair accuracy	Branch specificity	Expected
Connectivity oracle	1.000	1.000	1, 1
Window-only (radius 16)	0.500	0.500	exact ties
Size of query component	1.000	0.011	high, ≤ 0.5
Every $R = 16$ LocalCNN checkpoint	0.500	0.500	exact ties

Coverage and removal sizes. Of the positives in stream 8808, 89.7% have a qualifying cut site, giving 9,555 pairs; 5,863 fall in the conservative branch subset. The same rule qualifies 4,722 of the 5,256 test positives (89.8%). Their path lengths match the full positive set (median ℓ 69 versus 68). Using only those positives, against the same test negatives, changes $\mathrm{AUC}(\ell)$ by at most 0.014 at 64k (shortest bin 0.776 versus 0.789), 0.007 at 256k, and 0.001 at 1M, for initialization 0 of the full-access LocalCNN. The number of pixels a cut removes from the query's component has median 152 and mean 241 (range 2–2,090). Its median rises from 54.5 in the shortest path bin to 576.5 in the longest, with correlation 0.39 between ℓ and removal size.

Table 3: Minimal pairs by propagation depth for the five 1M full-access LocalCNNs. Medians with initialization ranges in brackets. Branch specificity uses the conservative subset. The bin $[31, 48)$ is the one inside the one-front budget of 32 together with the depths just above it; no positive in this pair set has $d_{\mathrm{prop}} < 31$. Beyond $d_{\mathrm{prop}} = 128$ only four pairs have a branch edit.

d_{prop}	pairs	pair accuracy	branch pairs	branch specificity
$[31, 48)$	5,124	0.999 [0.992, 1.000]	3,736	0.999 [0.983, 0.999]
$[48, 64)$	2,754	0.997 [0.984, 0.999]	1,570	0.995 [0.957, 0.997]
$[64, 80)$	999	0.991 [0.969, 0.998]	405	0.980 [0.941, 0.995]
$[80, 96)$	396	0.982 [0.967, 0.995]	112	0.973 [0.884, 0.982]
$[96, 128)$	233	0.961 [0.931, 0.974]	36	0.917 [0.778, 1.000]

B Seed-Position Reference and Path Bins

The seed-only scorer assigns each image the estimated $P(Y = 1)$ of its seed's border cell, from 500,000 images on unused stream 9901. Table 4 lists the 23 path bins used throughout, with the seed-only AUC in each. The five 64k $R = 32$ initializations have shortest-bin AUC 0.776, 0.770, 0.797, 0.757, and 0.779, all below the seed-only value of 0.831.

Table 4: Path bins on the test set and seed-only AUC against the common negative set.

bin	0	1	2	3	4	5	6	7	8	9	10	11
ℓ	37–46	47–49	50–51	52–54	55–56	57–58	59–60	61–62	63–64	65–66	67–68	69–70
positives	211	258	221	318	247	250	258	245	247	246	241	234
seed-only AUC	.831	.769	.734	.708	.664	.642	.579	.534	.533	.490	.480	.459
bin	12	13	14	15	16	17	18	19	20	21	22	
ℓ	71–72	73–75	76–78	79–82	83–86	87–91	92–99	100–108	109–123	124–157	158–265	
positives	210	249	235	226	210	216	219	222	208	200	85	
seed-only AUC	.467	.489	.505	.472	.471	.457	.497	.536	.511	.496	.496	

C Trajectory Details

At 5k updates, pair accuracy by path bin lies between 0.624 and 0.689 in run 0 and between 0.574 and 0.666 in run 1, with no trend in ℓ. At 10k, branch specificity in run 0 is 0.741 in the shortest bin, 0.61–0.67 in bins 18–21, and 0.579 in the final bin, which contains only 19 pairs. In run 1 the short and long groups are 0.721 and 0.730. At 50k, every bin with at least 79 pairs is between 0.94 and 1.00 in both runs, and the 19-pair final bin is 0.79 and 0.95. We therefore report no path-length order for connectivity specificity. Table 5 summarizes both runs at 5k resolution, Table 6 gives the 1k-resolution curves up to 15k, and Table 7 the regression. The plotted summary is the share $\beta_{\text{conn}}/(\beta_{\text{conn}} + \beta_{\text{size}} \log 153)$ of the predicted logit drop of the median cut (152 pixels removed), from the unrounded coefficients, defined only while $\beta_{\text{size}} > 0$. Through 4k updates both coefficients are close to zero, so the ratio's intervals are wide.

Table 5: Pair accuracy (PA) and branch specificity (B) during the two 1M runs.

updates	PA, run 0	B, run 0	PA, run 1	B, run 1
5k	0.648	0.524	0.620	0.519
10k	0.894	0.714	0.888	0.709
15k	0.941	0.834	0.932	0.824
20k	0.968	0.910	0.969	0.902
25k	0.978	0.955	0.983	0.953
30k	0.991	0.984	0.987	0.977
40k	0.994	0.991	0.993	0.992
50k	0.997	0.995	0.996	0.994

Table 6: Pair accuracy (PA) and branch specificity (B) from 1k to 15k updates, with 95% pair-bootstrap intervals.

updates	PA, run 0	B, run 0	PA, run 1	B, run 1
1k	0.503 [0.492, 0.512]	0.511 [0.499, 0.524]	0.497 [0.487, 0.507]	0.501 [0.489, 0.513]
2k	0.505 [0.496, 0.515]	0.499 [0.487, 0.512]	0.502 [0.494, 0.512]	0.497 [0.484, 0.509]
3k	0.497 [0.486, 0.505]	0.502 [0.489, 0.515]	0.501 [0.491, 0.511]	0.505 [0.493, 0.519]
4k	0.518 [0.509, 0.527]	0.510 [0.498, 0.521]	0.524 [0.515, 0.533]	0.505 [0.492, 0.517]
5k	0.648 [0.640, 0.656]	0.524 [0.512, 0.536]	0.620 [0.612, 0.629]	0.519 [0.508, 0.531]
6k	0.730 [0.723, 0.738]	0.549 [0.538, 0.561]	0.733 [0.726, 0.742]	0.545 [0.533, 0.558]
7k	0.787 [0.780, 0.794]	0.575 [0.563, 0.587]	0.795 [0.787, 0.801]	0.572 [0.560, 0.584]
8k	0.833 [0.825, 0.839]	0.626 [0.612, 0.637]	0.832 [0.825, 0.838]	0.624 [0.613, 0.635]
9k	0.853 [0.847, 0.860]	0.643 [0.631, 0.654]	0.853 [0.846, 0.859]	0.653 [0.640, 0.665]
10k	0.894 [0.888, 0.899]	0.714 [0.703, 0.725]	0.888 [0.883, 0.893]	0.709 [0.698, 0.720]
11k	0.894 [0.889, 0.900]	0.734 [0.722, 0.745]	0.902 [0.897, 0.908]	0.750 [0.740, 0.760]
12k	0.917 [0.912, 0.921]	0.782 [0.772, 0.794]	0.913 [0.908, 0.917]	0.777 [0.767, 0.787]
13k	0.928 [0.923, 0.933]	0.790 [0.781, 0.799]	0.927 [0.923, 0.932]	0.794 [0.784, 0.805]
14k	0.937 [0.933, 0.942]	0.827 [0.819, 0.837]	0.934 [0.930, 0.939]	0.809 [0.799, 0.819]
15k	0.941 [0.937, 0.945]	0.834 [0.824, 0.843]	0.932 [0.927, 0.937]	0.824 [0.815, 0.833]

Table 7: Per-checkpoint regression of the logit drop on $\log(1 + \text{pixels removed})$ and a disconnection indicator, for runs 0 / 1. The share is $\beta_{\text{conn}}/(\beta_{\text{conn}} + \beta_{\text{size}} \log 153)$ from the unrounded coefficients, with 95% pair-bootstrap intervals, and is defined only while $\beta_{\text{size}} > 0$. The ratio is shown for reference and is not plotted. In run 1 at 30k, $\beta_{\text{size}} = -0.052$ and $\beta_{\text{conn}} = 4.08$.

updates	ratio, run 0	ratio, run 1	β_{size}	β_{conn}	share
4k	4.0 [2.0, 7.7]	−0.2 [−2.4, 4.1]	≈ 0 / ≈ 0	≈ 0 / ≈ 0	—
5k	3.8 [2.3, 5.2]	4.7 [3.7, 5.7]	0.0004 / 0.0007	0.0017 / 0.0033	0.43 [0.30, 0.51] / 0.48 [0.42, 0.53]
6k	5.6 [5.0, 6.1]	6.0 [5.4, 6.6]	0.004 / 0.008	0.023 / 0.050	0.53 [0.50, 0.55] / 0.54 [0.52, 0.57]
8k	7.7 [7.3, 8.0]	7.4 [7.0, 7.8]	0.020 / 0.022	0.154 / 0.162	0.60 [0.59, 0.62] / 0.60 [0.58, 0.61]
10k	8.4 [8.0, 8.8]	8.9 [8.5, 9.4]	0.035 / 0.042	0.292 / 0.374	0.63 [0.62, 0.64] / 0.64 [0.63, 0.65]
12k	10.6 [10.2, 11.1]	10.6 [10.2, 11.1]	0.047 / 0.059	0.498 / 0.623	0.68 [0.67, 0.69] / 0.68 [0.67, 0.69]
15k	11.7 [11.2, 12.2]	11.6 [11.1, 12.1]	0.073 / 0.071	0.855 / 0.826	0.70 [0.69, 0.71] / 0.70 [0.69, 0.71]
20k	23.4 [22.0, 25.2]	15.7 [15.1, 16.5]	0.057 / 0.086	1.33 / 1.35	0.82 [0.82, 0.83] / 0.76 [0.75, 0.77]
25k	— ($\beta_{\text{size}} < 0$)	29.0 [27.2, 31.0]	−0.052 / 0.069	3.49 / 2.00	— / 0.85 [0.84, 0.86]

D Removal Size and Response Strength

Table 8: Pair accuracy at 256k within strata of pixels removed from the query's component. Medians over five initializations, with ranges in brackets; n is the number of pairs. Path groups are short ($\ell \leq 59$), middle ($60 \leq \ell \leq 99$), and long ($\ell \geq 100$).

pixels removed	n	all	short	middle	long
2–48	2,361	0.515	0.519 [0.493, 0.583]	0.516 [0.499, 0.532]	0.520 [0.480, 0.601]
48–152	2,416	0.722	0.804 [0.777, 0.898]	0.677 [0.657, 0.782]	0.684 [0.643, 0.773]
152–327	2,388	0.878	0.934 [0.923, 0.968]	0.874 [0.840, 0.936]	0.819 [0.792, 0.880]
327–2,090	2,390	0.923	0.972 [0.954, 0.993]	0.939 [0.904, 0.961]	0.882 [0.863, 0.915]

Table 9: Standardized cut effect (mean logit drop divided by the standard deviation of logits over test negatives) in selected path bins. Medians over five initializations, with ranges in brackets. At 64k the effect is below 1.3×10^{-5} in magnitude in every bin.

median ℓ in bin	256k	1M
44	0.053 [0.033, 0.068]	2.62 [1.15, 2.98]
56	0.043 [0.034, 0.066]	2.34 [0.92, 2.84]
63	0.027 [0.018, 0.048]	2.14 [0.80, 2.71]
72	0.028 [0.020, 0.048]	1.89 [0.68, 2.52]
85	0.020 [0.015, 0.036]	1.71 [0.58, 2.36]
104	0.011 [0.008, 0.017]	1.32 [0.40, 1.88]
138	0.009 [0.007, 0.011]	0.89 [0.25, 1.31]
171	0.0055 [0.0047, 0.0080]	0.41 [0.11, 0.60]

E Path-Conditioned AUC Views

This section collects the AUC-based analyses of the previous version. Where they make claims about which instances the model processes, the minimal pairs in the main text supersede them.

Trained local baseline. The $R = 16$ LocalCNN cannot observe the border seed or most of a connecting path, but it receives the seed coordinates and can learn correlations between the central window and the label. Increasing LocalCNN radius also changes depth and parameter count, so Δ_b is not a pure access comparison. Its curve at 64k closely follows the seed-only scorer (Table 4).

Horizon during training. With $H_{0.75}$ the furthest plotted bin with $\text{AUC}_b \geq 0.75$ (Section I), run 0 moves from 49 at 1k updates to 64 at 6k, 75 at 10k, 123 at 20k, 157 at 26k, and the final bin from 27k onward; run 1 is similar and reaches the final bin at 28k (Figure 8). Because the final bins cover a wide range of ℓ, the last jump looks more abrupt than the underlying curves.

Propagation depth with AUC. The high-performing LocalCNN has median AUC 0.991 for $d_{\text{prop}} \in [80, 82]$ and 0.969 for $d_{\text{prop}} \in [91, 95]$; among the 836 positives with $d_{\text{prop}} > 64$, the four

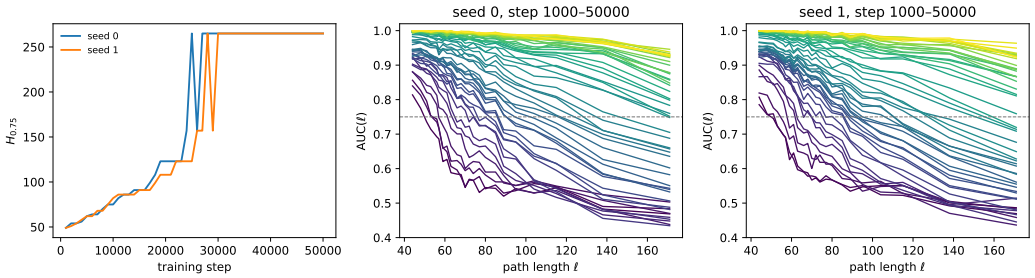


Figure 8: Path-conditioned AUC during the two 1M runs. Left: $H_{0.75}$. Middle and right: $\mathrm{AUC}(\ell)$ every 1k updates.

strong runs have positive error 0.21–0.31 and AUC 0.96–0.98 (Figure 9). Tied recurrence at $T = 64$ shows a stronger relation between performance and step count in two of three runs.

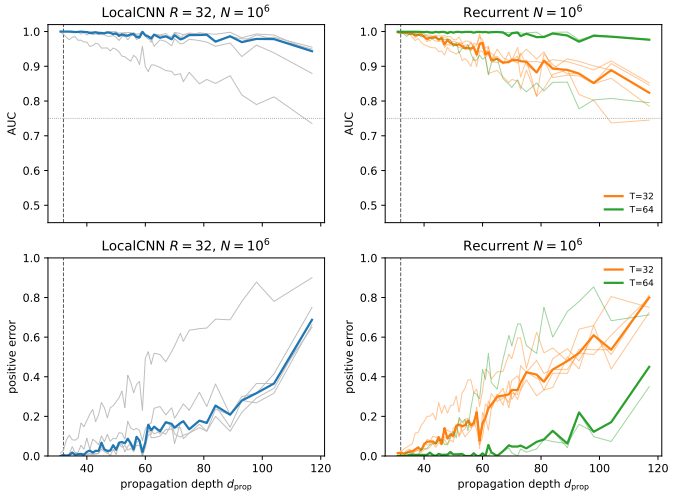


Figure 9: AUC and positive error by propagation depth at 1M. Left: LocalCNN $R = 32$. Right: tied recurrence; the $T = 64$ median is largely determined by two of three runs.

Architecture profiles and corridor density. At 256k, the excess over the $R = 16$ baseline is $+0.038$ in the shortest bin, $+0.109$ over $\ell \approx 59$–72, and -0.011 in the longest bin for the LocalCNN; the dilated $R = 32$ model keeps about $+0.17$ to $+0.20$ through the middle and tail; two partial U-Nets peak in the middle; and a partial tied recurrent model stays positive throughout (Figure 13). A LocalCNN given the recurrent model's zero-initialized residual map and hidden-state clamp stays close to the ordinary LocalCNN (Figure 13b). Within quartiles of occupancy in a narrow straight corridor between seed and query, AUC still declines with ℓ over $\ell \approx 53$–77 (Figure 12); the corridor does not follow the actual path, so this control is weak and is superseded by the minimal pairs.

F Model Sizes

Table 10: Parameter counts.

Model	Parameters
LocalCNN, L residual units	$385 + 41{,}088\,L$
$R = 16$	657,793
$R = 24$	986,497
$R = 32$	1,315,201
Dilated CNN, all radii	329,089
Tied recurrent CNN, all T	41,473
U-Net	7,760,385
Transformer, $d = 128$, 6 layers	1,714,817
Transformer, $d = 256$, 12 layers	10.5M
Four-neighbour MPNN	25,217

G Access and Optimization Budgets

Table 11: Executed and selected updates under validation loss early stopping (patience $4{,}000$ updates; evaluation every 200 updates; cap 50k unless noted). †Passes are derived as executed updates $\times\, 256/N$. "not recorded" means that the selected checkpoint of initialization 4 is not among the last 50 saved evaluations.

Run	executed updates	passes over pool†	selected update	stop	test TPR
LocalCNN R=32, 64k	5.0–5.4k	20–22	1.2–1.4k	early	≈ 0
LocalCNN R=32, 256k	12–14k	12–14	8.8–10k	early	0.04–0.21
LocalCNN R=32, 1M, initializations 0–3	47–50k	12–13	43–48k	early/cap	0.89–0.95
LocalCNN R=32, 1M, initialization 4	29.4k	7.5	not recorded	early	0.72
LocalCNN R=16, 1M	21–39k	5–10	17–35k	early	≈ 0
Trajectory runs, 1M, initializations 0–1	50k	12.8	49.6k, 49.8k	cap	0.94
Matched initialization LocalCNN, 256k	16.6k	16.6	12.6k	early	0.05
Recurrent T=32, 256k	42–50k	42–50	42–50k	early/cap	0.64–0.88
Recurrent T=32, 1M	49–50k	12.5–12.8	45–50k	early/cap	0.74–0.88
Recurrent T=64, 1M	15–30k	4–8	11–26k	early	0.82–0.98
Transformer, 1M (cap 100k)	4.6–7.4k	1.2–1.9	0.6–3.4k	early	0

H Cross Distribution Outcomes

Table 12: The same connectivity target under different input distributions. Prevalence, geometry, and executed training budgets differ, so these rows describe observed outcomes rather than rank intrinsic learnability. Evaluation on D4 models trained on Microsoft Common Objects in Context (COCO) masks can reuse training masks (Section P), while the Bernoulli trained transfer cannot.

Distribution	training	prevalence	majority error	LocalCNN R=32 error	TPR
D1 Bernoulli, $q = 0.60$	64k	0.263	0.263	0.264	0.000–0.005
D1 Bernoulli, $q = 0.60$	1M	0.263	0.263	0.017–0.112	0.72–0.95
D2 rooms, $c = 0.50$	64k–1M	0.042	0.042	0.042	0
D3 smooth shapes	64k	≈ 0.15	≈ 0.15	0.009 (mean)	≈ 0.97
D4 COCO masks, index overlap	64k	0.063	0.063	0.008–0.016	0.86–0.91
D1→D4 transfer	1M on D1	0.063	0.063	0	1.00

I Additional Path-Conditioned Results

AUC bins and horizon. The main AUC comparison and trajectories share 23 bins, distinct from the thresholded bins below. For plotted coordinates λ_b, $H_\tau = \max\{\lambda_b : \mathrm{AUC}_b \geq \tau\}$, without requiring earlier bins to pass; 265 labels the final bin.

I.1 Thresholded path profiles

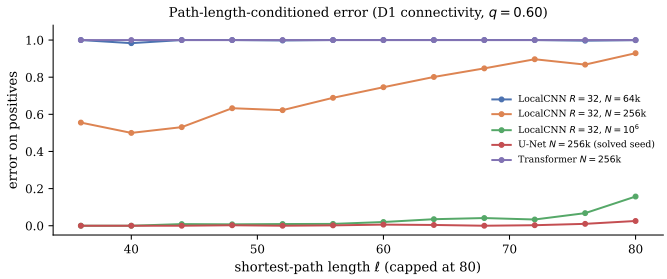


Figure 10: Positive error, or false negative rate, at the 0.5 threshold by shortest path length for single checkpoints: LocalCNN $R = 32$ at 64k, 256k, and 1M; a successful U-Net at 256k; and the Transformer at 256k. The first two bins contain 9 and 60 positives. The successful U-Net is nearly flat because it is already close to ceiling. This view motivates the path conditioned analysis but does not by itself establish an acquisition order.

I.2 Across initializations

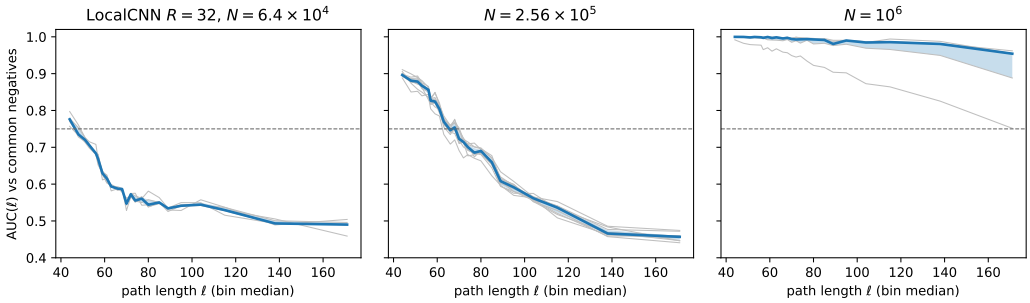


Figure 11: AUC(ℓ) for every LocalCNN $R = 32$ initialization (thin grey) with the median (blue) and interquartile band, at 64k, 256k, and 1M. The lower 1M curve is initialization 4, which stopped at 29.4k updates.

I.3 Corridor-density strata

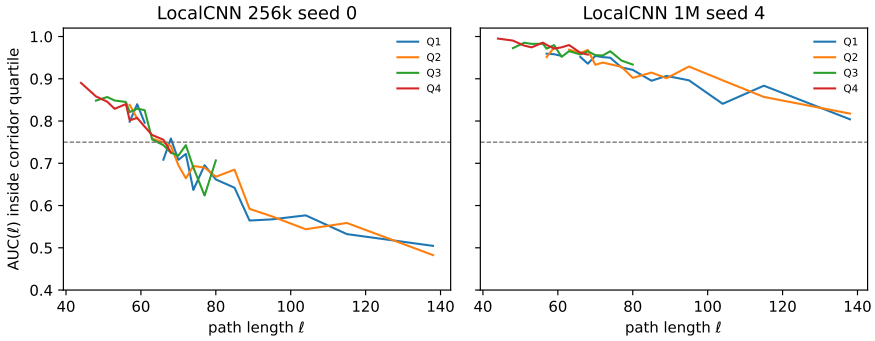
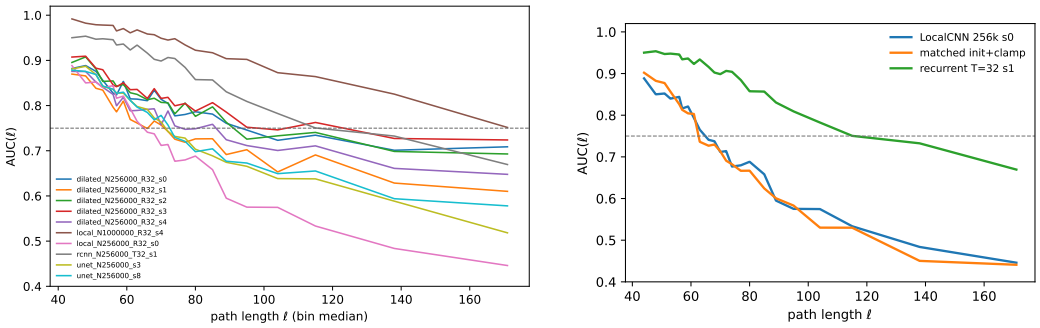


Figure 12: AUC(ℓ) within quartiles of seed–query corridor occupancy for the 256k initialization-0 checkpoint and the partial 1M initialization-4 checkpoint. Negatives are the common test-negative set. Within each quartile AUC declines with ℓ; curves stop where a quartile has fewer than 50 positives.

I.4 Architecture profiles and initialization control



(a) Partially trained checkpoints, raw AUC.

(b) Recurrent initialization control, 256k.

Figure 13: (a) Raw AUC(ℓ) for five dilated $R = 32$ runs, two partial U-Nets, and a partial tied recurrent run at 256k, together with the partial LocalCNN at 1M. (b) A LocalCNN using the recurrent model's zero initialized residual map and hidden state clamp remains close to the ordinary LocalCNN, while the recurrent model stays above both. Gradient clipping is shared.

I.5 Thresholded profiles, class-conditional context, and the exploratory horizon

Table 13 gives width-four thresholded profiles ending at 80+. Count-weighted rounded recalls give TPR 0.2057 at 256k and 0.9355 at 1M, consistent with initialization 0; the U-Net is 256k initialization 1 (TPR 0.990). The largest bin with positive error at most 0.5 is below 36 at 64k, near 40 at 256k, and at least 80+ at 1M. Threshold sensitivity, the 60-positive intermediate bin, and right censoring make this horizon exploratory.

Table 13: Thresholded positive error by path-length bin on the 5,256 test positives (single checkpoints; LocalCNN rows are $R = 32$, initialization 0).

model	36	40	44	48	52	56	60	64	68	72	76	80+
n in bin	9	60	226	395	437	508	500	483	459	358	295	1526
LocalCNN 64k	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
LocalCNN 256k	0.56	0.50	0.53	0.63	0.62	0.69	0.75	0.80	0.85	0.90	0.87	0.93
LocalCNN 1M	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.04	0.04	0.03	0.07	0.16
U-Net (successful, 256k)	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.03
Transformer 256k	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 14: Class conditional context for LocalCNN $R = 32$. †False positive rate (FPR) and precision use rounded natural test error e and TPR; $\mathrm{FPR} = (e - \pi(1 - \mathrm{TPR}))/(1 - \pi)$.

Pool	initialization	error	TPR	FPR†	precision†
256k	0	.270	.206	.083	.469
256k	1	.263	.149	.053	.499
256k	2	.266	.108	.043	.473
1M	0	.023	.937	.009	.975

J Propagation Budgets and Finite-Dependency Controls

Proposition 1 (Certified-propagation budget)**.** *Let G_X be the occupied four-neighbour graph. Suppose a procedure maintains certified reachable sets Q_t with $Q_0 = \{S\}$ and $Q_{t+1} \subseteq \{v : \exists u \in Q_t,\ d_{G_X}(u, v) \le k\}$. Then every $v \in Q_L$ satisfies $d_{G_X}(S, v) \le kL$.*

Proof. If v is added through $u \in Q_t$, then $d_{G_X}(S, v) \le d_{G_X}(S, u) + k \le k(t + 1)$; induct on t. □

Corollary 1 (Hop budget for 3×3 fronts)**.** *If Q_{t+1} contains only pixels joined to Q_t by a certified hop (a four-neighbour edge, or a diagonal step with an occupied shared corner), then every $v \in Q_L$ satisfies $d_{\mathrm{prop}}(S, v) \le L$.*

On the test set, 245 positives have $d_{\mathrm{prop}} \le 32$ (the minimum is 31, for 69 positives) and 5,011 have $d_{\mathrm{prop}} > 32$. A 32-layer front of this kind certifies at most the former; fronts grown from both endpoints need $d_{\mathrm{prop}} \le 2L$ before they can meet. The measured LocalCNN recall at $d_{\mathrm{prop}} \in [80, 82]$ and $[91, 95]$ exceeds both budgets. This excludes these two update classes, not every computation a 64-channel CNN can express: channels can encode boundary-connectivity summaries, compose information across the background, or carry statistical cues. Diagonal contact without an occupied corner does not certify a four-connected path, which is why d_{prop} admits diagonal hops only with a shared occupied corner.

Corollary 2 (Finite-dependency controls)**.** *For Chebyshev dilation or erosion of radius r, if the observation contains the operator input and uses the target's boundary convention, then $\epsilon^\star_R = 0$ for $R \ge r$. For four-connected reachability within K edges, every relevant pixel lies in the Manhattan ball of radius K, contained in the Chebyshev ball of radius K; with source and query observed, $\epsilon^\star_R = 0$ for $R \ge K$.*

Unrestricted connectivity has no radius bound independent of image size; at the canonical centre query, radius 32 contains the complete image.

K Seed-Level Outcomes and Aggregate Evaluations

Table 15: Primary three-initialization mean test error of the LocalCNN on D1 at $q = 0.60$ (36 configurations, 108 runs). †Mean TPR 0.15, balanced error 0.45. ‡Mean TPR 0.18, $R_{\mathrm{eff},95} = 15$; above the completion estimate 0.234. $^\#$Initializations 0–2: errors 0.023, 0.040, 0.020 (TPR 0.94, 0.89, 0.94; $R_{\mathrm{eff},95} \in \{25, 29\}$). With initializations 3 and 4 (0.017, 0.112), the five-run mean is 0.042 (TPR 0.89).

$N \setminus R$	4	8	12	16	24	32
10^3	0.263	0.263	0.263	0.263	0.263	0.263
$4{\cdot}10^3$	0.263	0.263	0.263	0.263	0.263	0.263
$1.6{\cdot}10^4$	0.263	0.263	0.263	0.263	0.263	0.263
$6.4{\cdot}10^4$	0.263	0.263	0.263	0.264	0.263	0.264
$2.56{\cdot}10^5$	0.263	0.263	0.263	0.263	0.265	$0.266^\dagger$
10^6	0.263	0.263	0.263	0.264	$0.259^\ddagger$	$0.028^\#$

Table 16: All LocalCNN $R = 32$ initializations. Initialization 4 at 1M stopped early at 29.4k updates.

64k			256k			1M		
init.	error	TPR	init.	error	TPR	init.	error	TPR
0	.263	.001	0	.270	.206	0	.023	.937
1	.263	.000	1	.263	.149	1	.040	.888
2	.265	.005	2	.266	.108	2	.020	.944
3	.263	.001	3	.262	.146	3	.017	.948
4	.264	.004	4	.270	.198	4	.112	.715
			5	.264	.104			
			6	.262	.121			
			7	.263	.036			
			8	.263	.164			
			9	.264	.111			

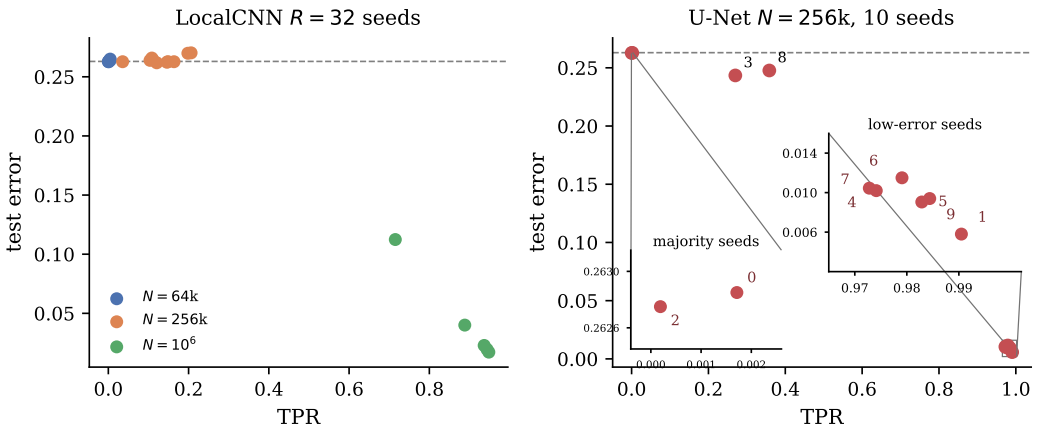


Figure 14: Seed-level outcomes (error versus TPR). Left: every LocalCNN $R = 32$ initialization at 64k, 256k, and 1M. Right: ten U-Net initializations at 256k. Strong runs form a cluster at high TPR and low error, separate from majority and partial runs; averages would mix them.

Table 17: U-Net outcomes. Initialization 2 is at the majority solution at both pools; the partial runs in Figure 13a are initializations 3 and 8 at 256k.

	U-Net 256k				U-Net 1M		
init.	error	TPR	status	init.	error	TPR	status
0	.263	.002	majority	0	.0043	.986	low error
1	.006	.990	low error	1	.0043	.992	low error
2	.263	.000	majority	2	.263	.001	majority
3	.244	.270	partial	3	.0043	.988	low error
4	.010	.974	low error	4	.0017	.996	low error
5	.009	.983	low error				
6	.012	.979	low error				
7	.010	.973	low error				
8	.248	.358	partial				
9	.009	.984	low error				

Table 18: Continuous outcomes at 1M. Ranges preserve failures as well as strong runs. Families are not matched for executed updates (Table 11); per-step FLOPs were not logged.

Model at 1M	runs	error range	TPR range	support
LocalCNN $R = 32$	5	.017–.112	.715–.948	full image
Dilated $R = 32$	5	.041–.051	.88–.92	full image
U-Net	5	.0017–.263	.001–.996	multiscale
Recurrent $T = 32$	5	.045–.083	.74–.88	radius 32
Recurrent $T = 64$	3	.008–.090	.82–.98	radius 64
Transformer	5	.2628	0	global
MPNN $T = 32$	5	.263	0	partial Manhattan

Threshold sensitivity. For concise reporting, a run could be called successful when error is below 0.05 and TPR exceeds 0.80; this is a reporting convention, not a definition of acquisition. Under it, the 1M counts are U-Net $4/5$, LocalCNN $4/5$, dilated $4/5$, recurrent $T = 32$ $1/5$, Transformer $0/5$, and MPNN $0/5$. A lenient rule (0.08, 0.75) gives $4/5$, $4/5$, $5/5$, $4/5$, $0/5$, $0/5$, and a strict rule (0.03, 0.85) gives $4/5$, $3/5$, $0/5$, $0/5$, $0/5$, $0/5$. The $1/5$ for recurrence reflects errors of 0.053–0.059 just above the cut, not an absence of discrimination. For scale, 95% Wilson intervals are $[0.38, 0.96]$ for $4/5$, $[0.31, 0.83]$ for $6/10$, $[0.04, 0.62]$ for $1/5$, and $[0, 0.43]$ for $0/5$; primary non-Transformer runs share one data stream. We therefore report continuous outcomes.

Balanced evaluation. A stream of 10k positives and 10k negatives drawn from the test seed is a different evaluation distribution. Majority checkpoints move from natural error 0.263 to balanced error 0.500; partial LocalCNN $R = 32$ runs at 256k have balanced error 0.443–0.469 (TPR 0.105–0.198); LocalCNN $R = 32$ at 1M has 0.032–0.062 (TPR 0.890–0.943); and a successful U-Net has 0.007 (TPR 0.991) (Figure 15, Table 19).

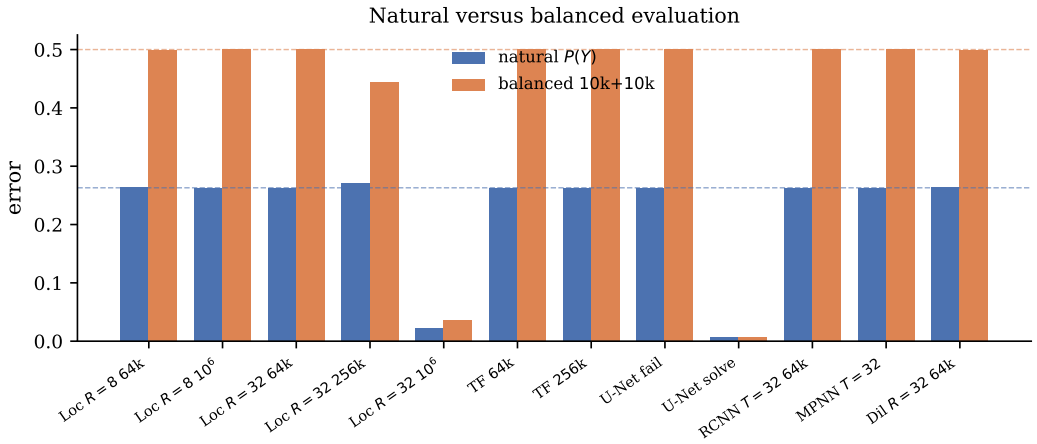


Figure 15: Natural versus class-balanced error on frozen checkpoints. Majority classifiers move from 0.263 to 0.50; successful LocalCNN (1M) and U-Net checkpoints stay near their natural error.

Table 19: Balanced evaluation of frozen checkpoints.

checkpoint	natural error	balanced error	balanced TPR
LocalCNN $R = 8$, 64k, init. 0–2	0.263	0.500	0.000–0.002
LocalCNN $R = 8$, 1M, init. 0	0.263	0.500	0.000
LocalCNN $R = 32$, 64k, init. 0	0.263	0.500	0.000
LocalCNN $R = 32$, 256k, init. 0–2	0.263–0.270	0.443–0.469	0.105–0.198
LocalCNN $R = 32$, 1M, init. 0–2	0.020–0.040	0.032–0.062	0.890–0.943
Transformer, 64k and 256k	0.263	0.500	0.000
U-Net, 256k, init. 0, 2	0.263	0.500	0.000
U-Net, 256k, init. 1	0.006	0.007	0.991
Recurrent $T = 32$, 64k	0.263	0.500	0.001
MPNN $T = 32$, 64k	0.263	0.500	0.000
Dilated $R = 32$, 64k	0.263	0.499	0.002

Matched pairs. On 200 opposite-label pairs that are identical within radius 8, LocalCNN $R = 8$, a dilated checkpoint, MPNN $T = 16$, and the Transformer agree on every pair (item accuracy 0.50). LocalCNN $R = 32$ at 256k agrees on 0.815 of pairs (accuracy 0.533); a partial U-Net at 64k agrees on 0.55 (accuracy 0.655); a successful U-Net at 256k disagrees on every pair (accuracy 1.0); MPNN $T = 32$ agrees on every pair. The pairs were not stratified by endpoint cluster size, bottleneck structure, or geodesic length, so they do not test the percolation-proxy explanation.

L COMMUNICATION INTERVENTIONS

Table 20: Fixed-parameter dilated CNN (eight residual units; schedules (1^8), $(1,1,1,2,2,2,3,3)$, $(1,1,2,2,3,3,5,5)$, $(1,1,2,4,4,4,8,8)$; 329,089 parameters). Individual $R = 32$ runs at 256k: errors 0.224, 0.244, 0.221, 0.215, 0.236 (TPR 0.46, 0.29, 0.39, 0.49, 0.37); at 1M: 0.044, 0.051, 0.041, 0.042, 0.044 (TPR 0.92, 0.88, 0.91, 0.88, 0.88).

N	R	runs	mean error [min, max]	mean TPR
$6.4{\cdot}10^4$	8/15/22/32	3 each	0.263	≤ 0.003
$2.56{\cdot}10^5$	8	5	0.263 [0.263, 0.263]	0.001
$2.56{\cdot}10^5$	15	3	0.263 [0.263, 0.264]	0.004
$2.56{\cdot}10^5$	22	3	0.263 [0.263, 0.263]	0.004
$2.56{\cdot}10^5$	32	5	0.228 [0.215, 0.244]	0.402
10^6	8	5	0.263 [0.263, 0.263]	0.000
10^6	15	3	0.265 [0.264, 0.265]	0.022
10^6	22	3	0.261 [0.260, 0.262]	0.166
10^6	32	5	0.044 [0.041, 0.051]	0.896

Table 21: Tied recurrent CNN (41,473 parameters). $T = 32$ at 256k: errors 0.053, 0.163, 0.049, 0.042, 0.061 (TPR 0.85, 0.64, 0.88, 0.88, 0.85; 42,400–50,000 updates). $T = 64$ at 256k: 0.007, 0.154, 0.029 (TPR 0.98, 0.64, 0.92). $T = 32$ at 1M: 0.045, 0.056, 0.053, 0.059, 0.083 (TPR 0.88, 0.85, 0.87, 0.86, 0.74). $T = 64$ at 1M: 0.011, 0.008, 0.090 (TPR 0.98, 0.98, 0.82).

N	T	runs	mean error [min, max]	mean TPR
$6.4{\cdot}10^4$	1–32	3 each	0.263	≤ 0.001
$6.4{\cdot}10^4$	64 (unstabilized)	3	0.612 [0.363, 0.737]	0.735
$6.4{\cdot}10^4$	64 (stabilized)	3	0.263 [0.263, 0.263]	0.000
$2.56{\cdot}10^5$	8	3	0.263 [0.263, 0.263]	0.000
$2.56{\cdot}10^5$	16	3	0.264 [0.263, 0.264]	0.012
$2.56{\cdot}10^5$	32	5	0.074 [0.042, 0.163]	0.820
$2.56{\cdot}10^5$	64	3	0.063 [0.007, 0.154]	0.846
10^6	8	3	0.263 [0.263, 0.263]	0.000
10^6	16	3	0.263 [0.263, 0.263]	0.003
10^6	32	5	0.059 [0.045, 0.083]	0.841
10^6	64	3	0.036 [0.008, 0.090]	0.928

Recurrent stabilization. The recurrent cell zero-initializes its residual 1×1 map and, for $T \geq 32$, clamps hidden states to $[-10, 10]$ after every step; gradient clipping at norm 1 is the trainer default for every run, recurrent and LocalCNN alike. The unstabilized 64k, $T = 64$ row is an earlier trio trained without these stabilizers; its initialization 0 has error 0.737 and TPR 1, a collapsed constant-positive predictor. The stabilized repeat sits at the majority solution. The matched LocalCNN control in Figure 13b uses the same zero-initialization, clamp, and clipping.

Table 22: Optimizer sweep at 1M, initialization 0. The sweep's primary-setting entries come from the sweep runs and differ slightly from the seed tables. A too-large learning rate collapses recurrent $T = 32$ to the majority solution.

model	setting	error	TPR
Dilated $R = 32$	AdamW 10^{-4}	0.084	0.794
Dilated $R = 32$	AdamW $3{\cdot}10^{-4}$ (primary)	0.041	0.919
Dilated $R = 32$	AdamW 10^{-3}	0.080	0.842
Dilated $R = 32$	SGD 10^{-3}	0.263	0.003
Recurrent $T = 32$	AdamW 10^{-4}	0.048	0.859
Recurrent $T = 32$	AdamW $3{\cdot}10^{-4}$	0.054	0.878
Recurrent $T = 32$	AdamW 10^{-3}	0.263	0.000
Recurrent $T = 64$	AdamW 10^{-4}	0.004	0.991
Transformer	AdamW 10^{-4}, $3{\cdot}10^{-4}$, 10^{-3}; SGD 10^{-3}	0.263	0.000

M Transformer and Partial-Support Controls

Three training streams (1101–1103) and three initializations at each of 256k and 1M give 18 Transformer replications, all at error 0.2628 and TPR 0 with $R_{\text{eff},95} \in \{31, 32\}$. Four optimizer settings (Table 22), three class-weighted BCE runs, and a 12-layer, $d = 256$ model (10.5M parameters) also select the constant-negative classifier. With $w_+ = (1 - \pi)/\pi \approx 2.81$, training alternates between constant-negative and constant-positive predictions, and early stopping retains a negative checkpoint (stopping at 4,200, 4,800, and 5,400 steps). For unweighted BCE, the best constant loss is binary entropy $H(\pi) = -\pi \ln \pi - (1 - \pi) \ln(1 - \pi) \approx 0.576$ nats; weighted BCE has a different constant optimum. On the path metric, the five 1M and three 256k seeds have short-bin AUC 0.42–0.47 and long-bin AUC 0.51–0.53. Raising the maximum cap to 100k or 200k leaves the stopping point near the original update count because early stopping remains active (Table 23, Figure 16); these are cap checks, not fixed-update training.

Table 23: Maximum-cap and capacity checks with early stopping active. The recurrent cap reruns are separate runs of the same configuration.

family	run	error	TPR	executed updates
Recurrent $T = 32$	original, init. 0	0.045	0.88	50,000
	100k / 200k cap	0.054	0.88	50,200
MPNN $T = 32$	original, init. 0	0.263	0	10,800
	100k / 200k cap	0.263	0	10,800
	width 256	0.263	0	10,000
Transformer	original, init. 0	0.263	0	7,400
	100k / 200k cap	0.263	0	7,200
	$d = 256$, 12 layers	0.263	0	7,000

Table 24: Other architectures at 64k (three initializations). The MPNN's Manhattan support misses much of the image at $T = 32$, so its failure is not a full-access result; its small effective radius at $T = 32$ suggests a gradient-flow limitation that we did not diagnose.

Model at 64k	mean error	TPR range	$R_{\text{eff},95}$	structural access
Grid MPNN ($T = 16$)	0.264	0.00–0.007	3	Manhattan radius 16
Grid MPNN ($T = 32$)	0.263	0.00	2	Manhattan radius 32 (partial)
U-Net	0.260	0.00, 0.00, 0.52	12, —, 30	multiscale
Transformer	0.263	0.00	32	global

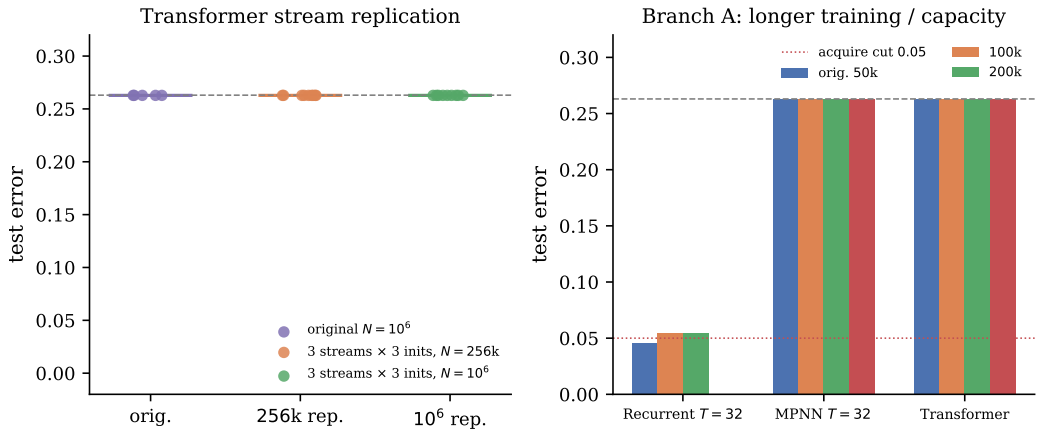


Figure 16: Left: the five 1M Transformer runs and the 18 stream×initialization replications, all on the majority line. Right: outcomes after raising maximum caps and after larger-capacity variants. Early stopping remains active, so larger caps do not mean more executed updates.

N Information Floors

Proposition 2 (Nested observations)**.** *If Z_{R_1} is a measurable function of Z_{R_2}, then $\epsilon^\star_{R_2} \le \epsilon^\star_{R_1}$. If Y is a measurable function of Z_R, then $\epsilon^\star_R = 0$.*

Proof. Any predictor of Z_{R_1} composed with the observation map is a predictor of Z_{R_2} with the same risk; take infima. A function determining Y attains zero error. □

These are standard Bayes-risk facts. For finite inputs, positive-mass completions with identical observations and opposite labels imply positive risk.

Estimator. For M anchors $z_m \sim P(Z_R)$ and K conditional completions each, $\widehat{p}_m = K^{-1}\sum_k Y_{mk}$ and $\hat{\epsilon}^\star_R = M^{-1}\sum_m \min\{\widehat{p}_m, 1-\widehat{p}_m\}$. Concavity gives downward finite-K bias; anchor-bootstrap intervals address neither this nor misspecified conditioning. Canonical D1 redraws are exact: unseen pixels remain independent given the visible patch and forced endpoints. Each radius uses its own anchors, so sampling variation can obscure population monotonicity. Common anchors can reduce between-radius variation, but finite completion estimates can still be nonmonotone.

Table 25: D1 completion estimates. Values at $R = 32$ are exact. At $q = 0.70$ the estimate is 0.126 at $R = 0$ and 0.108 at $R = 24$; at $q = 0.80$ and 0.90 positives dominate and low-radius estimates fall again.

R	0	8	16	24	32
$\hat{\epsilon}^\star_R$, $q = 0.60$	0.269	0.275	0.276	0.234	0
$\widehat{P}(Y=1)$, $q = 0.60$	0.269	0.275	0.278	0.273	0.271
$\hat{\epsilon}^\star_R$, $q = 0.65$	0.306	0.266	0.261	0.254	0
$\widehat{P}(Y=1)$, $q = 0.65$	0.693	0.703	0.706	0.707	0.707

At $q = 0.60$ the local window offers little improvement in zero–one risk through $R = 16$ (estimates close to $\widehat{P}(Y = 1)$), the estimate drops at $R = 24$, and it is exactly zero at $R = 32$. Near the percolation transition both labels occur and remote structure can flip the label, which is why $q = 0.60$ separates missing information from failure to use it. An estimate near the majority rate is compatible with AUC > 0.5: Bayes zero–one risk near the majority rate still allows ranking signal, as the $R = 16$ baseline shows. At $q = 0.60$, $R = 8$, the estimate varies between 0.271 and 0.277 across $M \in \{500, 1000, 2000\}$ and $K \in \{128, \ldots, 1024\}$ (Table 26).

Table 26: Stability of the D1 estimate at $q = 0.60$, $R = 8$. Bootstrap half-width at $M = 2000$ is about 0.005; the ambiguity fraction is 0.89–0.90.

M	$K = 128$	$K = 256$	$K = 512$	$K = 1024$
500	0.275	0.276	0.275	0.275
1000	0.275	0.277	0.274	0.276
2000	0.271	0.271	0.271	0.271

Table 27: Model–estimate uncertainty audit (model error as recorded for the audit; test and floor streams are independent finite samples with positive rates 0.2628 and ≈ 0.27–0.28). At $R = 8$ and 16 the difference is not separated from zero; at $R = 24$ the model stays above the estimate.

cell	model error	completion estimate	difference	95% interval
D1 q=0.60, R=8, N=10^6	0.263	0.269	−0.006	$[-0.013, +0.001]$
D1 q=0.60, R=16, N=10^6	0.263	0.270	−0.007	$[-0.014, +0.001]$
D1 q=0.60, R=24, N=10^6	0.263	0.234	+0.029	$[+0.016, +0.042]$

Table 28: Exact 4×4 (D0) references and matched-completion checks on task-specific slices (connectivity from $S = (0, 0)$ to $P = (1, 1)$ with both open). Agreement checks the implementation only when enumeration and Monte Carlo use the same input and query law.

Task slice	$R = 0$ exact / Monte Carlo	$R = 1$	$R = 2$	$R = 3$
Dilation $r = 1$	.00195 / .000	0 / 0	0 / 0	0 / 0
Erosion $r = 1$	.00195 / .00396	0 / 0	0 / 0	0 / 0
Connectivity (forced open)	.25000 / .24869	0 / 0	0 / 0	0 / 0
Holes (background-conditioned)	.08447 / .08384	.02197 / .01947	0 / 0	0 / 0

Table 29: D2 descriptive completion curves. Values below $R = 32$ come from freezing room and edge latents that intersect the patch; equivalence to conditioning on the raster alone is not established. The $R = 32$ value is exact.

R	0	4	8	16	32
$c = 0.50$	0.055	0.046	0.046	0.049	0
$c = 0.65$	0.133	0.109	0.108	0.107	0
$c = 0.80$	0.162	0.120	0.119	0.119	0

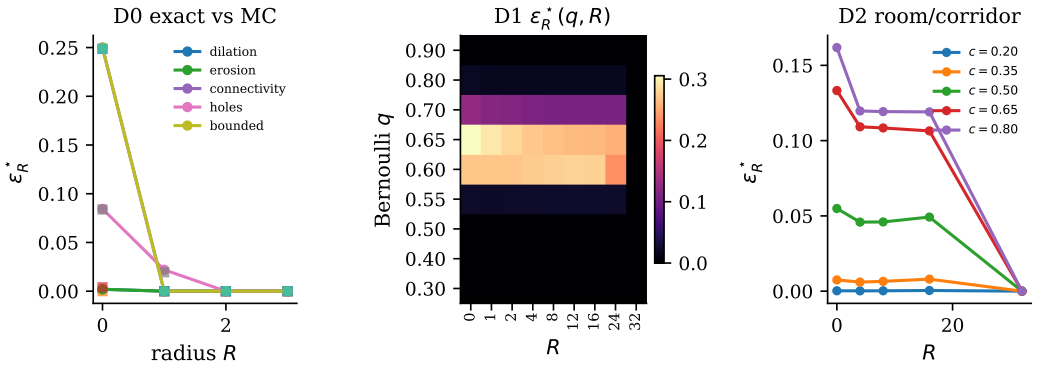


Figure 17: Information references. Left: exact small-grid values and completion checks. Centre: D1 estimates over occupancy and radius. Right: D2 estimates, descriptive below $R = 32$; the full-access value is exactly zero.

A 2×2 room graph (jitter 0, 16×16 canvas, seed $(0, 8)$, query $(8, 8)$, $c = 0.50$) has 16 edge configurations, all negative; enumeration and Monte Carlo ($M = 400$, $K = 64$) agree at $R \in \{0, 2, 4, 8\}$. Because it is label-degenerate, it checks only the implementation. For reconstruction at $R = 8$, model error 0.351 exceeds the descriptive marker-conditioned estimate 0.302 (difference 0.049, interval $[0.036, 0.062]$); marker conditioning is not established as exact. D3 and D4 receive no floor estimates.

O Other Generators and Controls

D2 rooms and corridors ($c = 0.50$). LocalCNNs at every tested (N, R), including five runs at $R = 32$ for each of 64k, 256k, and 1M, stay at error 0.042 with TPR 0 and $R_{\text{eff},95} = 4$ (Table 30, Figure 18), despite exact zero information risk at $R = 32$. With about 4.2% positives, imbalance and early stopping are plausible contributors alongside geometry. Evaluating the $c = 0.50$ graphs at 96^2, 128^2, and 192^2 (4,000 images each) leaves D2-trained $R = 8$ and $R = 32$ models at TPR 0, while prevalence falls from 0.042 to 0.027; fixed radii no longer give complete input at larger sizes. Width does not help at partial access: at D1 radius 8, widths 32–256 all stay at error 0.263.

Table 30: D2 LocalCNN outcomes at $c = 0.50$.

N	R	runs	mean error	mean TPR
$6.4{\cdot}10^4$	$4, 8, 16$	3 each	0.042	0.000
$6.4{\cdot}10^4$	32	5	0.042	0.000
$2.56{\cdot}10^5$	8	3	0.042	0.000
$2.56{\cdot}10^5$	32	5	0.042	0.000
10^6	8	3	0.042	0.000
10^6	32	5	0.042	0.000

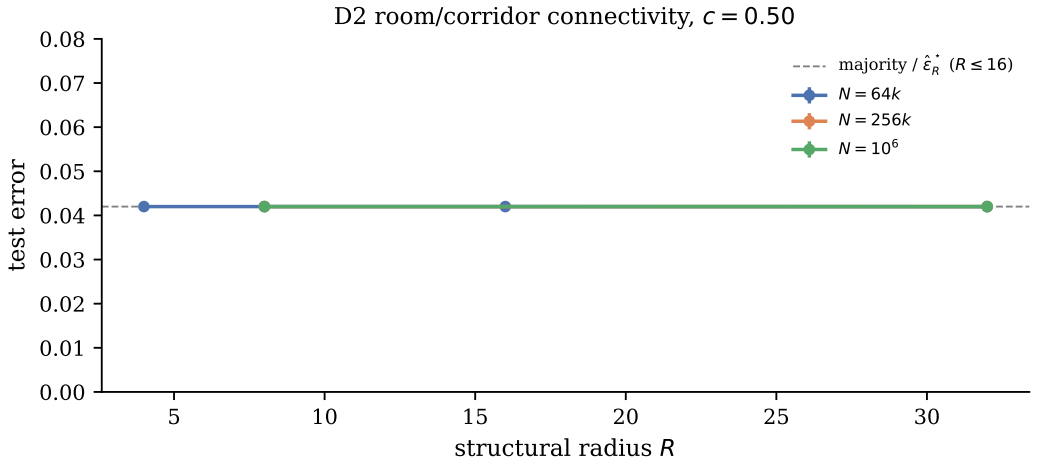


Figure 18: D2 at $c = 0.50$. Every LocalCNN cell, including full-access $R = 32$ at 1M, stays at the majority error 0.042.

D3 smooth shapes (prevalence ≈ 0.15). At 64k, LocalCNN mean error is 0.122 at $R = 8$ (TPR ≈ 0.48) and 0.009 at $R = 32$ (TPR ≈ 0.97). Bernoulli-trained LocalCNN $R = 8$ and the Transformer stay near the D3 majority rate; one Bernoulli U-Net transfers partially.

D4 COCO masks. At 64k (training prevalence ≈ 0.072), three D4-trained LocalCNN $R = 32$ seeds have test errors 0.008, 0.016, and 0.014 (TPR 0.91, 0.86, 0.88) on 4,000 images with positive fraction 0.06325; $R = 8$ stays at the majority rate, and U-Nets have errors of 0.005–0.006 (TPR 0.94–0.96). Because the mask is a function of the sample index (Section P), these D4-trained numbers can

reuse training masks. The Bernoulli-trained transfer cannot: LocalCNN $R = 32$ at 1M (initialization 0) has error 0, TPR 1, and true negative rate (TNR) 1 on the same 4,000 masks, so sample precision is 1; the 256k initialization 0 has error 0.109 (TPR 0.490); a successful Bernoulli U-Net has error 0.004 (TPR 0.93). Pool sizes differ, so this is finite-set transfer, not a matched comparison.

Morphological reconstruction. An erosion-marker cell at $q = 0.60$ has completion estimate ≈ 0.003 and is uninformative. A sweep over occupancy, marker size, and minimum marker distance selects the cell closest to balance ($q = 0.60$, 5×5 marker, distance ≥ 16, $P(Y = 1) = 0.5065$), with descriptive estimate 0.302 at $R = 8$ (interval $[0.292, 0.313]$) and exact zero at $R = 32$ (Table 31, Figure 19).

Table 31: Reconstruction, LocalCNN means over three initializations. At 1M, $R = 32$ runs have errors 0.027, 0.034, 0.029 (TPR 0.962, 0.948, 0.952).

N	$R = 8$ error (TPR)	$R = 32$ error (TPR)	$\hat{\epsilon}_8^\star$	$\epsilon_{32}^\star$
$6.4{\cdot}10^4$	0.385 (0.81)	0.365 (0.81)	0.302	0
$2.56{\cdot}10^5$	0.361 (0.84)	0.051 (0.92)	0.302	0
10^6	0.351 (0.83)	0.030 (0.95)	0.302	0

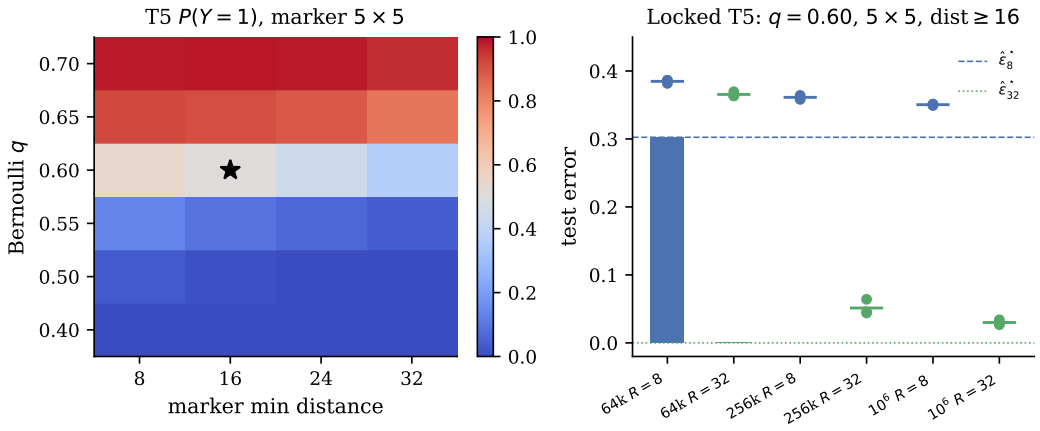


Figure 19: Reconstruction. Left: $P(Y = 1)$ over occupancy and minimum marker distance at marker size 5; the star marks the cell closest to $1/2$. Right: LocalCNN error by N and R; the $R = 8$ reference is descriptive, and the $R = 32$ floor is exactly zero.

Dense seeded-component prediction. The target is the seed's full component mask (Table 32). We report mean absolute prediction–target differences in counts of four-connected foreground components ($\Delta\beta_0$) and background holes not touching the border ($\Delta\beta_1$). A dense $R = 32$ LocalCNN gives most off-centre pixels only partial access, so the U-Net comparison mixes access, architecture, and supervision. LocalCNN $R = 8$ has higher exact-mask rate but far lower intersection over union than $R = 32$; small components may explain this, but we did not stratify by size.

Table 32: Dense seeded-component prediction, means over three initializations.

N	Model	pixel error	intersection over union	exact-mask	$\Delta\beta_0$ / $\Delta\beta_1$
$6.4{\cdot}10^4$	LocalCNN $R = 8$	0.132	0.07	0.27	1.4 / 65.2
$6.4{\cdot}10^4$	LocalCNN $R = 32$	0.075	0.51	0.12	15.2 / 36.8
$6.4{\cdot}10^4$	Recurrent $T = 32$	0.078	0.48	0.15	15.4 / 39.0
$6.4{\cdot}10^4$	U-Net	0.004	0.97	0.76	0.48 / 2.29
$2.56{\cdot}10^5$	LocalCNN $R = 8$	0.132	0.07	0.26	1.3 / 65.2
$2.56{\cdot}10^5$	LocalCNN $R = 32$	0.074	0.51	0.15	11.5 / 36.9
$2.56{\cdot}10^5$	Recurrent $T = 32$	0.078	0.45	0.32	3.3 / 39.9
$2.56{\cdot}10^5$	U-Net	0.002	0.985	0.82	0.34 / 1.07

Query position. Complete access is geometry-specific (Table 33, Figure 20). Inputs are recentred while labels are computed on the original image. At 64k, LocalCNNs at the geometry-specific full-access radii (44 for query $(20, 44)$; 55 for the worst sampled interior query) remain near the majority rate, as at the centre. Because even the centre model fails at 64k, these runs show access accounting, not position robustness of a successful predictor.

Table 33: Query-position controls at 64k.

geometry	R	$\hat{\epsilon}^\star_R$	LocalCNN error	TPR
centre $(32, 32)$	8	0.274	0.263	0.00
centre	32	0	0.264	0.00
off-centre $(20, 44)$	8	0.257	0.260	0.00
off-centre	32	0.115	—	—
off-centre	44	0	0.260	0.00
random interior	8	0.248	0.250	0.01
random interior	32	0.109	—	—
random interior	55	0	0.251	0.02

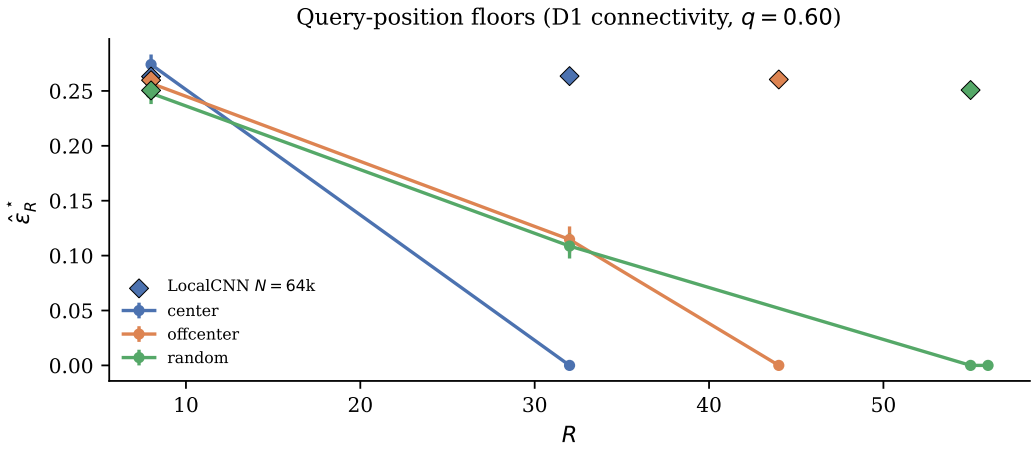


Figure 20: Completion estimates by query geometry at $q = 0.60$, with LocalCNN 64k means (diamonds). Even at radii that observe the complete input, the 64k models stay near the majority solution.

Other operators. Thresholding on uniform intensities is solved to within 0.50 ± 0.02 percentage points of error by a depth-2 LocalCNN ($N = 16{,}000$, three initializations). Radius-2 dilation at $q = 0.60$ is nearly label-degenerate and serves as a pipeline check; morphology at $q = 0.20$ reaches zero observed error at $R = 2$ and 8. Hole filling reaches error below 10^{-4} at $R \in \{4, 8, 16, 32\}$; a remote background route can change hole status, so we do not claim a zero local floor. Reachability

within $K = 8$ edges from a uniformly placed seed (positive rate ≈ 0.016) is beaten at every radius: mean errors 0.0075, 0.0065, and 0.0061 at $R = 4$, 8, 16 (TPR 0.67–0.72, 0.75–0.77, 0.66–0.79).

P GENERATORS, ORACLES, AND SPLITS

Inputs $X_i = G(\phi, s_{\text{split}}, i)$ are generated, never stored; ϕ collects generator parameters like q and c. Split seeds are $s_{\text{train}} = 1101$, $s_{\text{val}} = 2202$, $s_{\text{test}} = 3303$, $s_{\text{floor}} = 4404$, and out of distribution (OOD) $s_{\text{OOD}} = 5505$. Only Transformer replications use streams 1102 and 1103. The primary matrix uses stream 1101, initializations 0–2; extended cells reach 9.

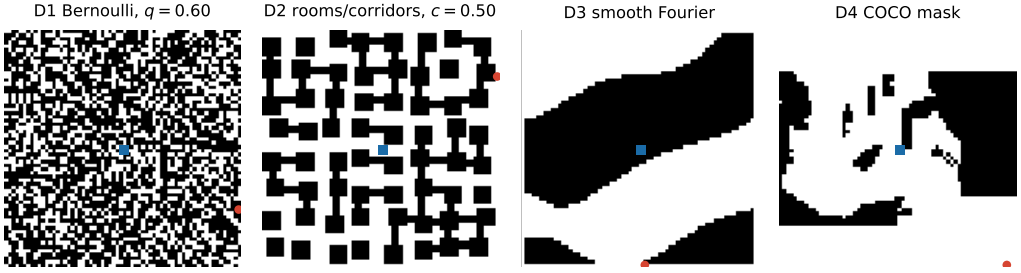


Figure 21: One test image from each generator (split seed 3303, index 0). The red dot is the border seed and the blue square is the centre query. **D1** is i.i.d. Bernoulli occupancy at $q = 0.60$. **D2** is the room/corridor layout at $c = 0.50$: 5×5 rooms in an 8×8 grid, joined by width-2 corridors. **D3** is a smooth Fourier field, thresholded. **D4** is one fitted COCO 2017 instance mask (Lin et al., 2014). D2 is defined here; it is not taken from an external room dataset.

D1. I.i.d. Bernoulli occupancy with $q \in \{0.30, 0.40, 0.50, 0.55, 0.60, 0.65, 0.70, 0.80, 0.90\}$ ($q = 0.60$ canonical); the seed is uniform on the border, the query is the centre, and both are forced open. Conditioned on the forced pixels and a visible patch, the remaining pixels are independent.

D2. An 8×8 latent room graph on a 64×64 canvas: 5×5 rooms in 8×8 cells, width-2 corridors between adjacent rooms with probability set by c, room jitter in $\{-1, 0, 1\}$ clipped to stay in-cell.

D3. $F(x, y) = \sum_{k=1}^{16} a_k \cos(\omega_{x,k} x + \omega_{y,k} y + \phi_k)$ on $[-1, 1]^2$ with $a_k \sim \mathcal{N}(0, 1/16)$, frequencies in $\{1, \ldots, 5\}$, and uniform phases, thresholded at a quantile and optionally closed.

D4. Built from COCO 2017 instance annotations (train and val), without image color values. Each image's instances are unioned, cropped to their bounding box, rescaled with nearest-neighbour interpolation to fit 64×64, and centred. Keeping fitted occupancy within a fixed band gives $n = 114{,}355$ masks. The generator takes entry $i \bmod n$, with deterministic flips when $i \geq n$. Overlapping indices reuse training masks at test time because masks ignore the split seed; Bernoulli-trained D4 evaluations are unaffected.

Oracles and audits. Connectivity uses four-connected component labelling, checked against breadth first search (BFS); ℓ counts edges. Holes, reconstruction, and morphology each use two implementations (component-based and flood/queue-based), and a batched graphics processing unit (GPU) flood fill was checked against component labelling. Secure Hash Algorithm 256 bit (SHA-256) audits of D1 prefixes (20k train, 8k validation, 20k test, 8k floor) and of D2 (8k train versus test) found no shared images, and every audited image was unique.

Q ARCHITECTURES, SIDE INFORMATION, AND HYPERPARAMETERS

LocalCNN: 1×1 projection to 64 channels, L residual units with one 3×3 convolution each, 1×1 centre readout; radius-limited models crop to $(2R + 1)^2$, and padding lies outside the centre's

structural field. **Dilated CNN**: eight residual units with the schedules in Table 20. **Recurrent CNN**: one residual unit applied T times, stabilized as in Section L. **U-Net**: widths 32, 64, 128, 256, two 3×3 convolutions per stage, skip concatenations, transposed-convolution upsampling. **Transformer**: $d = 128$, 6 layers, 8 heads, multilayer perceptron (MLP) ratio 4, learned positional encoding, one query token, global attention over pixel tokens. **Message passing neural network (MPNN):** two layer MLP message and update functions, mean aggregation over four neighbours; support after T steps is a Manhattan ball of radius T. Parameter counts for all families are in Section F. Gradient-support checks found zero gradient outside the stated support in all 15 tested configurations. $R_{\mathrm{eff},95}$ is the smallest Chebyshev radius about the query containing at least 95% of the absolute input-gradient mass of its logit, using the first image of one validation batch. It measures trained influence, not structural support.

Table 34: Shared side information. No family receives a privileged seed input.

family	input / readout	seed and query handling
Conv families, U-Net, MPNN	4 channels (occupancy, seed mask, s_y, s_x); centre logit	One-hot seed and two constant coordinate maps; LocalCNN crops to $(2R+1)^2$.
Transformer	same 4 channels as HW tokens plus a learned query token	Same seed maps; global attention with positional encoding.
Reconstruction	occupancy and marker; centre	No seed maps.
Thresholding	intensity; centre	No seed.

Hyperparameters. AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, batch 256, mixed precision, gradient clipping at global norm 1, no label smoothing, and no per-family search on the primary matrix. Validation BCE is evaluated every 200 updates with patience 20 evaluations; the cap is 80 epochs or 50k updates (100k–200k in the cap checks). Robustness runs use AdamW at 10^{-4} and 10^{-3} and stochastic gradient descent (SGD) with momentum 0.9. Weighted BCE is used only in the Transformer check. Per-step floating-point operations were not logged.