INTERACTION LOCALITY IN SPATIAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial reasoning models must combine location-bound computation with location-invariant structure. These roles are often read off state names, as when a recursive model's high-level state is called global and its low-level state local. However, a name does not fix how far an update's effect travels, and every task defines its own notion of nearby. To address this, we propose interaction locality, which reports the fraction of an intervention's causal effect, at a calibrated perturbation scale, that stays inside a relation declared in advance over the sites a task admits. We analyse HRM and TRM on Maze-Hard, Sudoku Extreme and ARC-AGI, compare them with a matched non-recursive Transformer, and extend the measurement to a 3D grounding model. In the recursive models, a perturbation of the low-level state lands more tightly in the high-level write it produces than in the next low-level call, an ordering that holds for every task-defined relation we test. Under the primary intervention, a size-matched control shows that ordering is propagation in four cells, retention at the perturbed site in one, and unresolved in the cell with the highest raw score. With added depth the non-recursive baseline and HRM keep spreading a perturbation, whereas the weight-tied recursive model plateaus. Because the score is also differentiable, using it as a fine-tuning regularizer raises measured locality without costing accuracy.

1 INTRODUCTION

Spatial reasoning couples local action to global structure: a route-following system depends on walls and immediate moves, and must keep those moves consistent with corridors and plans. Compact recursive architectures make the two levels explicit. The Hierarchical Reasoning Model (HRM; Wang et al. (2025)) and the Tiny Recursive Model (TRM; Jolicoeur-Martineau (2025)) solve Maze-Hard, Sudoku Extreme and ARC-AGI by repeatedly applying small Transformer modules (Vaswani et al., 2017) to a high-level state updated rarely and a low-level state updated often, and it is natural to read the high-level state as global planning and the low-level state as local refinement. However, "high" and "low" name a role and an update frequency; neither fixes how far a single update reaches in space, and global coordination can arise from local effects accumulating across iterations as readily as from one global update. Measuring reach directly is complicated by a second fact: every task defines its own notion of nearby: a Sudoku cell touches its box, a maze move reaches an adjacent cell unless a wall intervenes. A spatial reasoning task therefore comes with an interaction type, geometric constraints on where the model may act, and an objective that decides which relations matter, and reach must be measured conditional on all three.

Mechanistic interpretability supplies the tools: causal interventions localize mediators (Meng et al., 2022; Syed et al., 2023), and structured domains give an external geometry against which internal variables can be read (Toshniwal et al., 2022; Jenner et al., 2024; Ivanitskiy et al., 2023). Concurrent analyses of HRM and TRM characterize *what* their latent states converge to (Ren & Liu, 2026; Efstathiou & Balwani, 2026); what remains open is *how far* a single update's causal effect travels, measured against the task's own relation and without presupposing an H/L interface. There is also a quantitative trap: any score for the fraction of an effect landing near its source includes the source itself, so a high score can reflect concentration at the perturbed site instead of propagation.

We address this with *interaction locality*: for an intervention at one site, the fraction of the resulting causal effect that falls inside a relation declared in advance over the sites the task admits, read against the chance level those two choices imply. Declaring the relation from the task makes the measurement conditional and comparable at once, since the same intervention and control apply

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.

to every task; a size-matched random relation with the same chance level but no task structure is the control, and the declared relation's excess over it isolates task-aligned propagation from self-concentration, and we treat the control as part of the definition.

Our contributions are the following.

- **A measurement.** Interaction locality is an effect-mass fraction under a task-declared relation, read against a geometry-matched null and a size-matched control that separates task-aligned propagation from retention at the source; defined per measured edge, it applies to the recurrent edges of HRM and TRM and to the input→output edge of any sited model (Section 3).
- **What it finds in recursive models.** The high-level write is the more concentrated channel in every cell and for every relation tested, and under the primary intervention the control shows what that concentration is made of: propagation to other sites in four cells, retention at the source in HRM/Maze, and unresolved in TRM/ARC-AGI, the cell with the highest raw score (Sections 4.1 and 4.2). Reach depends on state, relation, edge and perturbation scale; with depth the baseline and HRM keep spreading while the weight-tied TRM plateaus, and at matched iteration count only TRM reorganizes its high-to-low balance (Section 4.3).
- **3D transfer and loss design.** The principle transfers to 3D grounding with the relation and the source convention changed, finding causal locality from both directions of the intervention and at the grounded output (Section 4.4). The score is also differentiable, so fine-tuning against it raises measured within-H locality without costing accuracy, mostly by retention at the perturbed site (Section 5).

2 Background and related work

Recursive reasoning models. HRM (Wang et al., 2025) and TRM (Jolicoeur-Martineau, 2025) apply small Transformer modules (Vaswani et al., 2017) repeatedly to a persistent state under an adaptive-computation wrapper (Graves, 2016); within one step a low-level state is updated several times and a high-level state is written once and carried forward. HRM uses separate modules for the two roles and TRM one weight-tied module, so comparing them varies parameter sharing at a fixed schedule. Both are trained on Maze-Hard, Sudoku Extreme and ARC-AGI (Chollet, 2019), whose geometry is known independently of the model. The architecture fixes the schedule and the sharing, and nothing about how far one update reaches.

Causal probes and what they estimate. Activation patching localizes causal mediators by replacing an internal state and reading the change downstream (Meng et al., 2022; Syed et al., 2023); attribution methods trace influence through gradients (Baehrens et al., 2010; Simonyan et al., 2013). Jacobians describe an infinitesimal neighborhood (Rifai et al., 2011; Liu et al., 2026) and sparse autoencoders decompose activations into features (Elhage et al., 2021; Bricken et al., 2023; Lieberum et al., 2023). These methods locate where an effect is mediated; none says how much of it stays within a task-defined relation, what fraction chance would give, or whether it stayed at the intervened site. Interaction locality adds those three quantities to a finite intervention. Finite interventions can also push a state off the model's natural activation distribution (Grant et al., 2026); those studied there substitute a vector computed from other data, whereas ours adds isotropic noise, and we measure the displacement it produces (Appendix H.1); a substitution drawn from the model's own states at the same site leaves the raw ordering unchanged (Table 8). That confound is distinct from self-mass, and the control of Section 3.1 addresses only the latter.

Spatial structure in trained models. Probing work has recovered board state in chess (Toshniwal et al., 2022), look-ahead and planning in game-playing networks (Jenner et al., 2024; Taufeeque et al., 2024), spatial variables in maze-solving Transformers (Ivanitskiy et al., 2023; 2024), and geographic structure in language models (Spies et al., 2024; Sabbata et al., 2025; Tigges et al., 2024). Embodied and vision-language systems raise the same question at scale (Chen et al., 2024; Zheng et al., 2024; Kim et al., 2024; Kang et al., 2026; Wu et al., 2025); MTU3D (Zhu et al., 2025) grounds language in reconstructed scenes (Dai et al., 2017), a metric geometry in place of a grid. This literature establishes that spatial variables are present and decodable with instruments tailored

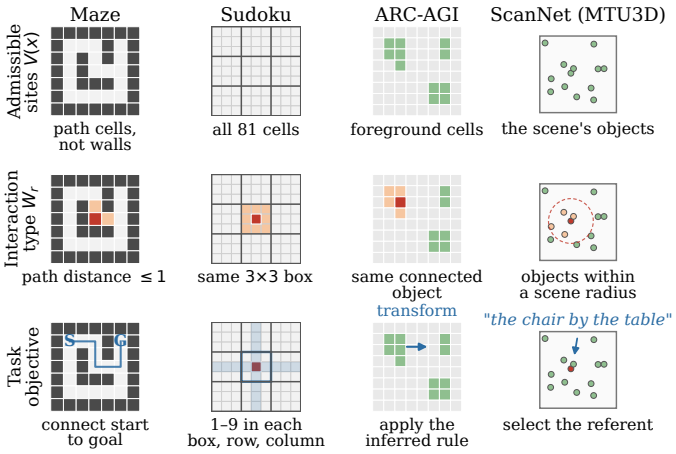


Figure 1: The three components of a spatial reasoning task across the four settings. **Admissible sites** $V(x)$ (top) are where the task lets the model act; an **interaction type** W_r (middle) is a relation declared from task structure, with a source v in red and its neighborhood $N_r(v)$, which always contains v, in orange; the **objective** (bottom) decides which relations are worth declaring. The null b_r follows from $V(x)$ and W_r alone.

to each domain. It leaves open how far the causal effect of changing one of them travels, and it offers no single procedure under which a grid puzzle, a non-recursive baseline and a metric 3D scene can be compared. Interaction locality supplies that procedure; between settings only the task's relation changes, and for the 3D functional the source convention.

Dynamics of recurrent computation. Recurrent computation has long been characterized through fixed points (Sussillo & Barak, 2013; Pascanu et al., 2013; Gu & Dao, 2024; Paulo et al., 2024), and two concurrent studies apply that lens to the models we analyze, reporting attractor-like dynamics and questioning whether either model reasons in the sense its framing suggests (Ren & Liu, 2026; Efstathiou & Balwani, 2026). Those accounts characterize what the latent state converges to, and the spatial reach of an update is a separate question; the two are orthogonal, since a structured guess may be spatially local or global, and they meet at convergence, where the settling they describe appears in our depth profiles as a plateau (Section 4.3).

3 Framework and Experimental Design

3.1 Interaction Locality

Sites and admissible sites. An input x presents a finite set of *sites* at which the model reads or writes, such as grid cells or object slots. The task's constraints pick out which the model may act on; that subset is the *admissible sites* $V(x)$ (Figure 1, top row), over which every quantity below is defined.

Interaction types. An *interaction type* is a symmetric binary relation $W_r : V(x) \times V(x) \to \{0, 1\}$ with $W_r(v, v) = 1$, indexed by r, declared from the task's structure before any model response is inspected (Figure 1, middle row). For a source v, $N_r(v) = \{u \in V(x) : W_r(u, v) = 1\}$ is its neighborhood and $n_r(v) = |N_r(v)|$ its size.

Measured edges and effect mass. A *measured edge* $a \to b$ names the state a where an intervention is injected and the state b where it is read. For source and target $v, u \in V(x)$, let $z_b^{\text{clean}}(u)$ and $z_b^{\text{pert}}(u)$ be the vector at site u of state b before and after v is perturbed at a. The *effect mass* at u caused by v is $A_{a\to b}(u, v) = \delta(z_b^{\text{pert}}(u), z_b^{\text{clean}}(u))$, where $\delta \geq 0$ is a response magnitude, the ℓ_2 distance in the primary analysis.

The score. *Interaction locality* is the fraction of a source's total effect mass that lands inside its declared neighborhood,

$$L^{(r)}_{a\to b}(v) \;=\; \frac{\sum_{u\in V(x)} W_r(u,v)\, A_{a\to b}(u,v)}{\sum_{u\in V(x)} A_{a\to b}(u,v)}, \tag{1}$$

averaged over admissible sources and over examples wherever the denominator is nonzero. Higher values mean more concentrated influence, and the score is conditional on the edge, the type and the admissible sites.

The geometry-matched null. Neighborhoods differ in size across tasks by an order of magnitude, so a raw $L^{(r)}$ is read against chance for the same geometry. Under uniform assignment over admissible sites a source v places a fraction $n_r(v)/|V(x)|$ of its mass inside $N_r(v)$, and the *geometry-matched null* $b_r(x) = |V(x)|^{-1}\sum_v n_r(v)/|V(x)|$ is that fraction averaged over sources; it follows from $V(x)$ and W_r alone and lets a 3×3 Sudoku box and a four-connected maze cell each be read against its own chance level.

Self-concentration and the size-matched control. $A_{a\to b}(v,v)$ contributes to the numerator of Equation (1) for every type r. Write $s_{a\to b}(v) = A_{a\to b}(v,v)/\sum_{u\in V(x)} A_{a\to b}(u,v)$ for the *self-mass share*, the fraction of effect that stays at the perturbed site. A high $L^{(r)}$ can reflect a large s as easily as propagation along task structure, and the null does not separate them; a control that keeps neighborhood size and source but discards task structure does. For each source v, draw a random relation $\widetilde{W}_v$ containing v and $n_r(v)-1$ further sites chosen uniformly from $V(x)\setminus\{v\}$, compute Equation (1) with it, and average over $M = 100$ draws. The result is the *size-matched control* $L^{\mathrm{ctrl}}_{a\to b}(v)$, and $L^{(r)}_{a\to b}(v) - L^{\mathrm{ctrl}}_{a\to b}(v)$ is the *excess* of the declared type over chance-with-self. By linearity of expectation (Appendix A.2),

$$\mathbb{E}\big[L^{\mathrm{ctrl}}_{a\to b}(v)\big] \;=\; s_{a\to b}(v) \;+\; \big(1 - s_{a\to b}(v)\big)\,\frac{n_r(v)-1}{|V(x)|-1}, \tag{2}$$

so a positive excess is what remains once source concentration and neighborhood size are both accounted for, and the declared type and its control share the null. When $n_r(v)$ is a small fraction of $|V(x)|$, the control is nearly $s_{a\to b}(v)$ itself and most of a raw value is self. Excluding the source from both sums of Equation (1) gives the *off-diagonal* locality, the direct reading of propagation to other sites. We report raw score, control, excess and off-diagonal difference together in Section 4.

Paired comparisons. Channels measured on the same example are compared by their paired difference, $\Delta(x) = L_{\text{within-H}}(x) - L_{\text{within-L}}(x)$, averaged over examples with a percentile-bootstrap interval, which uses the between-channel covariance that separate marginal intervals discard.

3.2 Edges, readouts and calibration

Notation for the recurrence. Within one adaptive-computation step a recursive model runs P phases; phase p applies the low-level module C times, producing $z_L^{(p,1)},\dots,z_L^{(p,C)}$, then writes $z_H^{(p)}$ from $z_L^{(p,C)}$ and $z_H^{(p-1)}$. Both are indexed by site; HRM has $P=2$, $C=2$ with distinct modules, TRM $P=3$, $C=4$ (6 on Sudoku) with one module in both roles, called $C+1$ times per phase.

Measured edges. Three edges live on the recurrent trajectory (Figure 3, Appendix A.1). **Within-L** perturbs $z_L^{(p,c)}(v)$ and reads $z_L^{(p,c+1)}$; **within-H** perturbs $z_L^{(p,C)}(v)$ and reads the write $z_H^{(p)}$ it produces; **cross-HH** perturbs $z_H^{(p)}(v)$ and reads the carry $z_H^{(p+1)}$. These need a recurrent state; the fourth, **input→output**, perturbs the input embedding of v and reads the output logits, and is the edge shared with the non-recursive baseline and MTU3D.

Readouts. Changing δ reads the same edge at three levels with $V(x)$, W_r and b_r fixed. *State locality*, the primary analysis, uses the ℓ_2 distance between hidden vectors. With $\ell(u)$ the output logit vector at site u, *output-effect locality* substitutes $A^{\mathrm{out}}(u,v) = \|\ell_v^{\mathrm{pert}}(u) - \ell^{\mathrm{clean}}(u)\|_2$. *Prediction-flip locality* uses the indicator $F(u,v) = \mathbb{1}[\arg\max_c \ell^{\mathrm{pert}}_{v,c}(u) \neq \arg\max_c \ell^{\mathrm{clean}}_c(u)]$, the fraction of changed predictions inside $N_r(v)$ conditioned on at least one change; the unconditional flip rate is a magnitude, not a locality, and is reported separately.

Table 1: Finite-noise patching locality, 500 examples per cell. $\Delta = L_{\text{within-H}} - L_{\text{within-L}}$ is computed per example with a percentile-bootstrap 95% CI; every interval excludes zero. Values are raw and so include the effect at the perturbed site; Table 2 removes it.

Architecture	Task	Baseline	within-L	within-H	Δ [95% CI]
HRM	Maze	.026	.266	.344	$+.078$ [$+.075$, $+.082$]
HRM	Sudoku	.111	.332	.374	$+.043$ [$+.042$, $+.044$]
HRM	ARC-AGI	.367	.477	.577	$+.100$ [$+.092$, $+.108$]
TRM	Maze	.026	.032	.263	$+.230$ [$+.228$, $+.232$]
TRM	Sudoku	.111	.113	.213	$+.100$ [$+.099$, $+.101$]
TRM	ARC-AGI	.379	.627	.688	$+.061$ [$+.055$, $+.067$]

Finite-noise activation patching and calibration. The primary intervention adds Gaussian noise to one site, $z_a(v) \leftarrow z_a(v) + \sigma_a\,\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, with σ_a fixed per model and edge on a held-out probe set before any locality is computed. On Maze and Sudoku it is calibrated toward a 30% *self-drop* target, the rate at which the perturbed site's own clean-correct prediction changes; on ARC-AGI, where per-cell predictions are too sparse, the signal-to-noise ratio is set to one, so two models are perturbed at equal behavioral strength. Because every block output is RMS-normalized to $\|z\| = \sqrt{d}$, with $d = 512$ the hidden width, σ_a reads as a noise-to-signal ratio: the SNR target gives $\sigma_a = 1$, while self-drop yields 1.9 to 49, largest on the least noise-sensitive low-level states. The next block consumes the inflated state before its own post-norm restores $\sqrt{d}$ (Appendix H.1); the achieved self-drop misses the target in five of twelve analogs, between .23 and .62 (Table 29), and the ordering is re-read at six scales in Appendix D.1. Where a model's noise sensitivity is itself changed by an intervention, as in Section 5, the scale is shared across the compared arms instead.

3.3 Models and comparisons

Recursive models. We analyze released HRM and TRM checkpoints on Maze-Hard, Sudoku Extreme and ARC-AGI at the converged phase and call (Appendix C.1), 500 accepted examples per cell, ARC-AGI examples requiring two to fifteen foreground components.

The non-recursive control. To ask what recursion contributes we train an eight-layer single-phase Transformer on Maze-Hard in the same codebase and wrapper and measure it on the input→output edge, the only edge defined for it; because the baseline plateaus below TRM, by 1.8 points in path-cell accuracy and from $.73$ to $.49$ in exact match, we report the comparison across the baseline's training trajectory.

The 3D model. MTU3D (Zhu et al., 2025) grounds a language query in reconstructed indoor scenes (Dai et al., 2017). Its sites are object slots, its admissible set the scene's objects, and its interaction type metric. Each change follows from the domain: a scan has no grid, so the neighborhood becomes a per-scene radius calibrated to hold about seven objects, which keeps its share of the scene comparable across rooms; and the intervention deletes an object's input, whose own slot change measures the deletion, so the source is excluded from both sums of Equation (1) and the control coincides with the null (Corollary 3). The design is nested: recursion decides which edges exist, the task decides the interaction type, and the same score and null apply throughout, the 3D functional in the source-excluded form of Section 3.1.

4 Measuring interaction locality

4.1 The ordering, raw and off-diagonal

The anchor. We anchor each cell at the phase and call at which the ℓ_2 change between successive states has plateaued (Figure 4); Sections 4.1 and 4.2 read there and Section 4.3 relaxes it.

The raw ordering. Within-L and within-H share an injection site and differ only in what is read, so their paired difference isolates the write. In all six cells (Table 1) a perturbation of the low-level

Table 2: Declared interaction types against a size-matched control. The *excess* is the paired Δ of the declared type less that of its control, 100 random relations matched to the type's neighborhood size per example, which share the null of Section 3.1; **bold** marks an excess that clears the control, and $|\mathrm{sd}|$ gives it in control standard deviations, since a permutation p is pinned at its floor beyond about five (Appendix D.2). *Self* is the within-H self-mass share. Δ^{off} is the paired difference with the source excluded from both sums, with a bootstrap interval.

Cell	Type	Δ	control	excess	$\lvert\mathrm{sd}\rvert$	self	Δ^{off} [95% CI]
HRM/Maze	path	.077	.108	$-.031$	87.8	.291	$-.0303\ [-.0310, -.0294]$
TRM/Maze	path	.143	.117	$+\mathbf{.026}$	98.6	.131	$+.0299\ [+.0296, +.0303]$
HRM/Sudoku	box	.120	.031	$+\mathbf{.089}$	44.6	.172	$+.1098\ [+.1091, +.1105]$
	row	.079	.031	$+\mathbf{.048}$	24.1		$+.0622\ [+.0613, +.0630]$
	column	.037	.031	$+\mathbf{.007}$	3.4		$+.0098\ [+.0092, +.0103]$
TRM/Sudoku	box	.135	.068	$+\mathbf{.067}$	41.0	.089	$+.0738\ [+.0730, +.0745]$
	row	.125	.068	$+\mathbf{.057}$	35.0		$+.0630\ [+.0623, +.0636]$
	column	.100	.068	$+\mathbf{.032}$	19.5		$+.0350\ [+.0346, +.0354]$
HRM/ARC	component	.080	.032	$+\mathbf{.048}$	47.5	.345	$+.0508\ [+.0451, +.0568]$
	color	.060	.030	$+\mathbf{.030}$	29.9		$+.0356\ [+.0287, +.0423]$
	radius	.075	.041	$+\mathbf{.034}$	35.9		$+.0431\ [+.0380, +.0483]$
TRM/ARC	component	.036	.044	$-.008$	10.4	.523	$-.0043\ [-.0088, +.0002]$
	color	.028	.034	$-.006$	7.7		$-.0043\ [-.0092, +.0009]$
	radius	.031	.046	$-.015$	19.5		$-.0142\ [-.0180, -.0103]$

state lands more tightly in the high-level write it produces than in the next low-level call, with a paired interval excluding zero. The pattern is largest for TRM on Maze, where within-L sits at the geometry-matched null (.032 against .026) while within-H is .263. Cross-cycle carry is less concentrated than the write in every cell, consistent with local writes being integrated across iterations, and the ordering holds under every task-defined relation tested (Table 2, column Δ; Appendix D.1). Calibration, however, changes the picture. Concretely, the ordering holds in 16 of 18 self-drop checks and 10 of 18 matched-SNR checks, and at every scale only in the ARC-AGI cells (Appendix D.1). Both self-drop exceptions are HRM/Sudoku, whose within-L scale in Table 1 overshoots its target, a self-drop of .615 against .30; at the lower scale the sweep fitted for the same target, the cell reverses. SNR keeps the perturbed state within 0.3 of the natural between-example spread but changes the output on 0.5–5% of examples, whereas self-drop displaces it by 0.8 to 5.3 times that spread (Appendix H.1), so the ordering is strongest where the intervention is furthest from the activation distribution. Resample patching, which substitutes the same site's state from another example, injects a state the model produces there and needs no scale; it changes the source's prediction in 9 to 61% of examples, and the raw ordering holds in all six cells, both Maze cells included ($+.133$ and $+.160$, Table 8). The ordering therefore persists under activation resampling as well as the primary Gaussian-noise intervention.

With the source excluded. How much of that ordering is propagation to other sites is a separate question, since every raw value includes the effect at the perturbed site. The off-diagonal paired difference of Table 2 removes it. HRM/Maze is a resolved reversal at $-.0303\ [-.0310, -.0294]$, so its raw Δ of $+.077$ is carried by the source; TRM/ARC-AGI is unresolvable at $-.0043$ $[-.0088, +.0002]$ against the highest raw within-H in the suite, .688. The high-level write is therefore the more concentrated channel in all six cells; in four that concentration is interaction with other sites, in HRM/Maze it is retention at the source, and in TRM/ARC-AGI the two cannot be separated on the component type (its radius type reverses at $-.0142\ [-.0180, -.0103]$). Under resample patching the off-diagonal difference also reverses in TRM/Maze and TRM/ARC-AGI (Table 8), so which cells propagate to other sites depends on the intervention family.

4.2 Task structure versus self-concentration

The control. A declared type should beat a relation of the same size that the task does not name. Table 2 tests this for the fourteen drawable type–cell combinations: ten clear the control, the four that do not are HRM/Maze and all three TRM/ARC-AGI types, and one clears weakly, HRM's

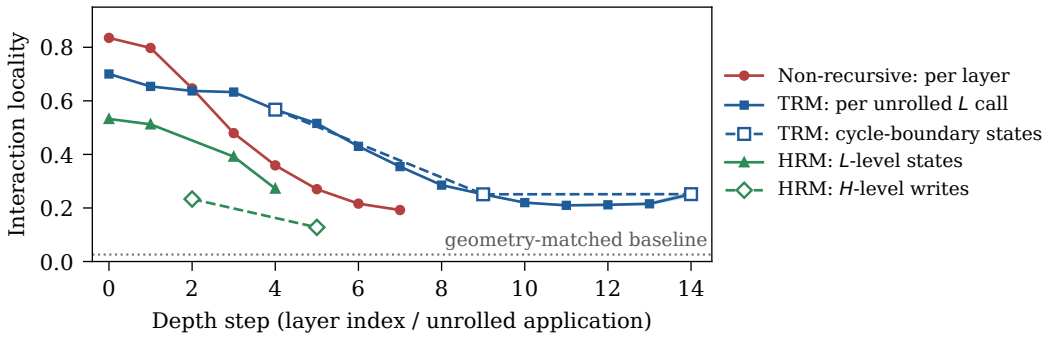


Figure 2: How causal reach develops with computational depth on the shared input→output edge (Maze, $n = 100$ matched examples). Color denotes model, line style role: solid for the baseline's layers, TRM's unrolled low-level applications and HRM's low-level states; dashed with open markers for TRM's cycle-boundary states, which lie on its own curve, and HRM's high-level writes. 95% bootstrap CIs over examples, drawn for the solid series, are mostly narrower than the lines (mean width .006, at most .023). A depth step is one layer for the baseline, two for TRM and four for HRM, so only the shape of each curve is comparable across models.

Sudoku column at $+.007$. In all fourteen rows the sign of the excess agrees with that of Δ^{off}, so the control is the raw ordering with self removed; the $|\text{sd}|$ column separates rows a floored p cannot, and the one row within five standard deviations, HRM's Sudoku column, resolves at $p = .0030$ with $M = 1000$ (Appendix D.2).

Why the exceptions fail, and why that is predictable. Equation (2) gives each control exactly in expectation from two quantities, the self-mass share and the neighborhood size (Proposition 2), and each failure is one of those quantities dominating. On Maze the path type relates a cell to three of roughly a hundred sites, so the control is almost entirely self and scores a Δ of .108 for HRM against the declared .077; on TRM/ARC-AGI the self-mass share is .523, so most of the numerator is the source. Neither says the model is not local; both say the raw score misattributed source concentration to task structure.

4.3 Trajectory with computational depth

Only the shared input→output edge can say what recursion contributes, so we perturb a source cell's embedding there and read after each depth step for the baseline, TRM and HRM on matched Maze examples.

All three globalize; only one stops. Every model in Figure 2 starts local and spreads, and they end differently: the baseline is still spreading when its layers run out, at $-.0389$ $[-.0406, -.0372]$ per layer, HRM at $-.0299$, while TRM has stopped, with a flat final-cycle fit of $-.0011$ $[-.0024, +.0002]$ and a cycle-boundary state that does not move between its last two cycles ($+.0005$ $[-.0096, +.0112]$).[1] On the same edge at matched task, TRM/Maze is more local than HRM/Maze, $7.7\times$ its null against $4.6\times$; the Sudoku cells are near $2\times$ and both ARC-AGI cells sit at their null (Table 31). Matched at six cycles by re-entering the carry, all three converge, the baseline included, so state re-entry suffices for convergence; what separates them is the balance they converge to, which reorganizes only in TRM (Table 11, Appendix E.1).

Reaching behavior. A score on hidden states leaves open whether the measured spread is task-relevant; reading the same intervention at the output settles it. Taking the paired within-H minus within-L difference at the hidden state, at the logit change and at the prediction flip of Section 3.2, all eighteen intervals exclude zero, so the ordering holds at every readout in every cell, and the separation is larger at the output than in the hidden state in five of six cells, the opposite of what signal diffusion would produce (Table 9). The flip readout is the independent one, since the position-wise

[1] A per-layer reading (Figure 16) gives $+.0066$ $[+.0048, +.0085]$, so TRM re-concentrates at layer granularity while the module fit is flat; "stopped spreading" holds under both.

head makes the logit readout a weighted reading of the same effect. Frequency and concentration dissociate: unconditional flip rates run from .004 to .294, the largest in TRM/Sudoku's within-L channel, whose off-diagonal locality is the lowest in the suite.

4.4 3D GEOMETRY

Locality is present and reaches the decision. Zero-ablating one object's input features gives locality .524 [.497, .552] at the visual-to-grounding handoff against a null of .420, above the null in every stratum, checkpoint and encoder depth, and grounding-logit locality is .513 against .434. The ablation also changes which object the model grounds in 11.6% of scenes, a magnitude and not a locality.

The comparable axis. Whether the 3D model settles with depth depends on which axis is read, and two exist. Reading the input ablation at successive encoder layers gives a profile that plateaus above the null; perturbing one object's populated query slot at layer one and reading later layers, the analogue of Figure 2's edge, gives a profile that decays at $-.0136$ $[-.0152, -.0121]$ per layer against a flat layer-zero control (Table 23, Appendix G.1). We rest on the second, because at layer zero every object's query slot holds the same vector, so an input-level ablation acts before any object-specific representation exists, and we report both. On the matched axis MTU3D spreads with depth as the non-recursive 2D model does, which places it with depth and removes an apparent counterexample to Section 4.3.[2]

The same structure from the opposite direction. A real locality should not depend on which way the intervention runs; ablation asks where a corruption spreads, patching a clean layer-k state into the corrupted run what a site's final state depends on. On the same 200 scenes and null, recovery locality is .589, .609 and .649 at injection layers one to three (Table 24), so locality is not an artifact of readout direction. We read that probe as a level and score the recovered mass unnormalized, since a per-target denominator is distance-dependent (Appendix G.2).

5 OPTIMIZING INTERACTION LOCALITY

The score of Equation (1) is differentiable in the model parameters once the interaction type is fixed, so its complement can be added to the task loss. On TRM's final high-level write $z_H \leftarrow f(z_H, z_L)$, with f its shared module, the one edge that carries gradients, we add calibrated noise to $z_L(v)$ for a sampled source v, recompute the write, and penalize the fraction of the change landing outside $N_r(v)$,

$$\mathcal{L}_{\text{loc}} = \mathbb{E}_v \left[\frac{\sum_u \big(1 - W_r(u,v)\big)\, \rho_v(u)}{\sum_u \rho_v(u) + \eta} \right], \qquad \rho_v(u) = \big\| z_H^{\text{pert}(v)}(u) - z_H(u) \big\|_2, \tag{3}$$
$$\mathcal{L} = \mathcal{L}_{\text{task}} + \beta\, \mathcal{L}_{\text{loc}},$$

where u ranges over the admissible sites, $\eta > 0$ guards the denominator and $\beta \geq 0$ sets the strength. This is the training-time counterpart of the within-H measurement, on the same edge, under the same interaction type and at the same calibrated scale.

The score moves, and accuracy holds. Fine-tuning Maze-TRM under Equation (3) against an identical $\beta = 0$ control, three seeds per arm (Table 3), within-H locality is monotone in β, from .272 at the control to .308 at $\beta = 0.1$; the control and strongest arm have disjoint seed ranges, $[.270, .275]$ against $[.305, .313]$; cell accuracy holds between .9800 and .9833; and within-L is unchanged at .032 in every arm, the signature of an objective acting on the high-level write alone. All arms are read at the shared σ at which the loss was applied, since a per-checkpoint refit moves the measuring stick with the model (Table 3, right column).

[2] Only the sign and the shape of the two profiles are comparable. Their levels are not, since the 2D baseline starts at roughly 32× its geometry-matched null and MTU3D at 1.15× its own.

Table 3: Locality-aware fine-tuning of Maze-TRM, three seeds per arm, $n = 200$, sixty epochs from the released checkpoint; intervals are the lowest and highest value over seeds. Within-H rises with β while within-L and accuracy hold; the shared-σ columns are primary.

	Shared σ (primary)			Per-model σ
Variant	within-H	within-L	Cell acc.	within-H
Released checkpoint	0.264	0.032	0.9831	0.262
Fine-tuned, $\beta = 0$ (control)	0.272 [0.270, 0.275]	0.032	0.9833	0.290 [0.178, 0.412]
Fine-tuned, $\beta = 0.01$	0.279 [0.278, 0.282]	0.032	0.9800	0.312 [0.279, 0.340]
Fine-tuned, $\beta = 0.1$	0.308 [0.305, 0.313]	0.032	0.9824	0.310 [0.194, 0.394]

What the result supports. The effect is modest, about 13% over the control, with no claim about a downstream capability. Raising the self-mass share would satisfy the regularizer as surely as propagation would (Section 4.2), and splitting the score by Proposition 1 shows it does most of the work: the self-mass share accounts for 95% of the rise to $\beta = 0.1$ (.214 to .259), and off-diagonal locality rises from .062 to .066, resolved across seeds (Table 6). The gain is mostly retention at the perturbed site, with a smaller gain in propagation.

6 Discussion

From a label to a relation. Interaction locality turns *H is global* from a description of a state into a measurement answered jointly by a state, an interaction type, a perturbation scale and a recursive time edge, none of which the architecture fixes. The informal story is thereby revised: the high-level write is the concentrated channel and the carry the broad one, a pattern consistent with local writes being integrated across recursive time. HRM's writes sit low in Figure 2 only because that edge reads accumulated reach and HRM schedules its high-level block at the bottom of the within-cycle decay, its own dispersion being no larger than a low-level block's.

What the control changes. The most consequential result is methodological. A mass-fraction locality score includes the perturbed site in every neighborhood, and Equation (2) makes the consequence exact: the smaller the declared relation, the more of a raw score is self, and two of the six headline cells lose their ordering once that term is removed. Any measurement of where an intervention's effect lands inherits this arithmetic. What recursion changes is where reach settles, which agrees with the attractor accounts of Section 2 and adds the spatial reach of the settled state, which they leave open.

From measurement to design. Because the score is differentiable, what describes a trained model can be asked of it during training (Section 5). Open next are how interaction locality emerges over training, whether it predicts robustness or transfer, and whether a regularizer on off-diagonal locality avoids the retention that dominates the raw one.

7 Limitations

We compare released checkpoints with no accuracy-matched sweep, so the claims concern mechanisms in available models, not accuracy; TRM/Maze is the one cell we trained, the public checkpoint then available being undertrained. The recursion comparison is Maze only, with a baseline 1.8 points below TRM and two recursive models that sample recursion unevenly, so only signs and scale-invariant ratios should be read across models. Every locality value is conditional on a declared interaction type and admissible set, and magnitudes depend on the calibration target, so we rest the claims on orderings and the control. No calibrated intervention here is both on-distribution and behaviorally consequential: self-drop reaches its target at 1.9 to 49 times the state norm, SNR stays near the distribution but changes the output on 0.5–5% of examples, the self-drop target is missed in five of twelve analogs, and at the embedding edge the injected vector exceeds the vocabulary's diameter (Appendix H.1), which the control does not address; resample patching is both at the perturbed site but moves the downstream state beyond the natural spread in both Maze cells (Table 8). The fine-tuning demonstration is one task, one architecture and three seeds.

REPRODUCIBILITY STATEMENT

An anonymized code release accompanies the submission (`https://anonymous.4open.science/r/TinyMechInterp-6D42/`); it records, for every measurement it produces, the checkpoint identity and step, the perturbation scale per model and edge, the achieved sample size and the random seeds, and every table in this paper is generated from those records by the included scripts. The score, null and control are defined in Section 3.1, with the interaction types and admissible sites fixed in configuration before any model response is inspected; the edges, readouts and calibration procedure are in Section 3.2, and checkpoint provenance and anchors in Appendix C.1. Statistical conventions are stated once in Section 3.1 and used throughout, and where a result depends on a convention both settings are reported, as in the two σ columns of Table 3. The size-matched control's draw count and null-matching assertion are given in Section 3.1 and Appendix D.2.

AI USE STATEMENT

The authors used large language models as general-purpose assistive tools, covering drafting and editing prose, writing and debugging analysis and plotting code, and preparing figures and tables. All experimental design choices, interventions, statistical procedures and claims are the authors' own, and every number reported in this paper was produced by the released code and verified by the authors against its output. The authors take full responsibility for the contents of the paper, including any remaining errors.

ETHICS STATEMENT

This work analyzes publicly released research checkpoints on puzzle benchmarks and on ScanNet, an existing dataset of reconstructed indoor scans. No new data were collected and no human subjects were involved. Interaction locality is intended as an auditing tool, and we expect its main effect to be making compact recursive and embodied spatial systems more inspectable before deployment in safety-relevant settings such as navigation and manipulation. The same analysis could inform stronger spatial reasoning systems, and interpretability results can be over-trusted, so an error in a locality measurement or misplaced confidence in a diagnostic could contribute to unsafe navigation, manipulation failures or unwarranted trust in an embodied agent. We therefore present interaction locality as a complement to task-specific safety and robustness evaluation, and not as a substitute for it.

REFERENCES

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(61):1803–1831, 2010. URL https://jmlr.org/papers/v11/baehrens10a.html.

Trenton Bricken et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024. doi: 10.1109/CVPR52733.2024.01370. URL https://arxiv.org/abs/2401.12168.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

A. Efstathiou and A. Balwani. Recursive reasoning as attractor landscape search: Mechanistic dynamics of the tiny recursive model. In *Workshop on Latent and Implicit Thinking: Going Beyond CoT Reasoning*, 2026. URL https://openreview.net/forum?id=kKps9W1K7n.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Satchel Grant, Simon Jerome Han, Alexa R. Tartaglini, and Christopher Potts. Addressing divergent representations from causal interventions on neural networks. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2511.04638. Oral.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Michael Ivanitskiy, Alexander F. Spies, Tilman Räuker, Guillaume Corlouer, Christopher Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia Diniz Behn, Katsumi Inoue, and Samy Wu Fung. Linearly structured world representations in maze-solving transformers. In *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*, volume 243 of *Proceedings of Machine Learning Research*, pp. 133–143. PMLR, 2024. URL https://proceedings.mlr.press/v243/ivanitskiy24a.html.

Michael I. Ivanitskiy et al. Structured world representations in maze-solving transformers. *arXiv preprint arXiv:2312.02566*, 2023.

Erik Jenner et al. Evidence of learned look-ahead in a chess-playing neural network. In *Advances in Neural Information Processing Systems*, 2024.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*, 2025.

Raphaela Kang, Hongqiao Chen, Georgia Gkioxari, and Pietro Perona. Linear mechanisms for spatiotemporal reasoning in vision language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2601.12626.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Tom Lieberum et al. Does circuit analysis interpretability scale? Evidence from multiple choice capabilities in Chinchilla. *arXiv preprint arXiv:2307.09458*, 2023.

Toni J. B. Liu, Baran Zadeoğlu, Nicolas Boullé, Raphaël Sarfati, and Christopher J. Earls. Jacobian scopes: Token-level causal attributions in llms. *arXiv preprint arXiv:2601.16407*, 2026. URL https://arxiv.org/abs/2601.16407.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. ROME: Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1310–1318. PMLR, 2013. URL https://proceedings.mlr.press/v28/pascanu13.html.

Gonçalo Paulo, Thomas Marshall, and Nora Belrose. Does transformer interpretability transfer to RNNs? *arXiv preprint arXiv:2404.05971*, 2024.

Zirui Ren and Ziming Liu. Are your reasoning models reasoning or guessing? a mechanistic analysis of hierarchical reasoning models. *arXiv preprint arXiv:2601.10679*, 2026. URL https://arxiv.org/abs/2601.10679.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 833–840, 2011. URL https://icml.cc/2011/papers/455_icmlpaper.pdf.

Stef De Sabbata, Stefano Mizzaro, and Kevin Roitero. Geospatial mechanistic interpretability of large language models. *arXiv preprint arXiv:2505.03368*, 2025.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. URL https://arxiv.org/abs/1312.6034.

Alex F. Spies, William Edwards, Michael I. Ivanitskiy, Adrians Skapars, Tilman Räuker, Katsumi Inoue, Alessandra Russo, and Murray Shanahan. Transformers use causal world models in maze-solving tasks. *arXiv preprint arXiv:2412.11867*, 2024.

David Sussillo and Omri Barak. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, 25(3):626–649, 2013. doi: 10.1162/NECO_a_00409.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*, 2023. URL https://arxiv.org/abs/2310.10348.

Mohammad Taufeeque, Philip Quirke, Maximilian Li, Chris Cundy, Aaron David Tucker, Adam Gleave, and Adrià Garriga-Alonso. Planning in a recurrent neural network that plays Sokoban. *arXiv preprint arXiv:2407.15421*, 2024.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. Llm circuit analyses are consistent across training and scale. *arXiv preprint arXiv:2407.10827*, 2024.

Shubham Toshniwal et al. Chess as a testbed for language model state tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*, 2025.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *arXiv preprint arXiv:2506.09965*, 2025. URL https://arxiv.org/abs/2506.09965.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. doi: 10.1109/CVPR52733.2024.01293. URL https://arxiv.org/abs/2312.02010.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, Siyuan Huang, and Qing Li. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. *International Conference on Computer Vision (ICCV)*, 2025. URL https://mtu3d.github.io/.

A Framework details

A.1 The measured edges on each schedule

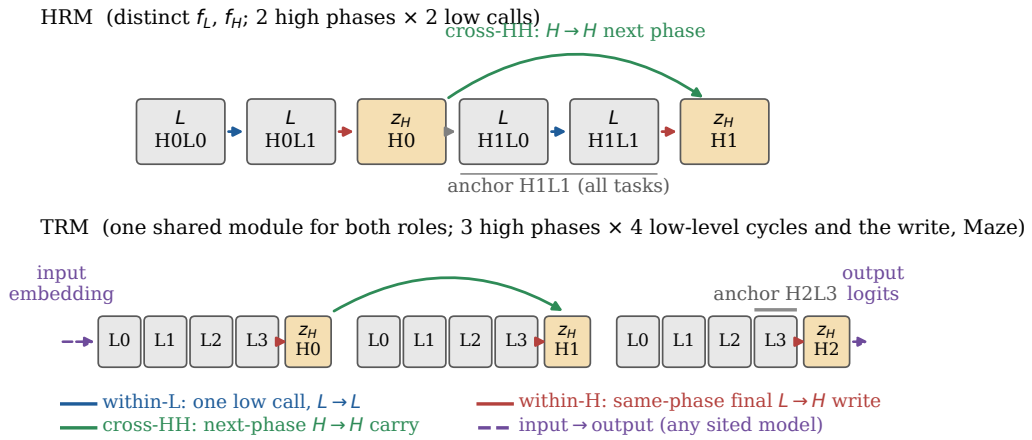


Figure 3: The recurrent schedules and the four measured edges, defined in the text. HRM alternates distinct low-level and high-level modules over two phases; TRM applies one shared module in both roles over three phases, so its edges are state-role and time edges and imply no separate parameters. The TRM row omits within-L arrows between individual low-level calls for legibility; the grey arrow is the carry of the high-level state into the next phase, and grey underlines mark the converged anchors at which the main results are read.

A.2 Formal statements: decomposition of the score

The statements below hold for one measured edge $a \to b$, one declared type r and one source $v \in V(x)$, and we drop those indices where no ambiguity arises. Write $A(u) = A_{a\to b}(u, v)$, $T = \sum_{u\in V(x)} A(u) > 0$ for the total effect mass, $s = A(v)/T$ for the self-mass share, $n = n_r(v)$

and $N = |V(x)|$. When $s < 1$, the off-diagonal locality is

$$L_{\text{off}} = \frac{\sum_{u \neq v} W_r(u, v)\, A(u)}{\sum_{u \neq v} A(u)},$$

the score of Equation (1) with the source removed from both sums.

Proposition 1 (Decomposition)**.** *If $s < 1$, then $L^{(r)}(v) = s + (1 - s)\, L_{\text{off}}$. If $s = 1$, then $L^{(r)}(v) = 1$.*

Proof. Since $W_r(v, v) = 1$, the numerator of Equation (1) is $A(v) + \sum_{u \neq v} W_r(u, v) A(u)$. Dividing by T gives s plus $\big(\sum_{u \neq v} A(u)/T\big) L_{\text{off}} = (1 - s) L_{\text{off}}$. If $s = 1$, every off-source term vanishes and the ratio is $A(v)/T = 1$. □

Proposition 2 (Size-matched control)**.** *Let $\widetilde{W}_v$ contain v and $n - 1$ further sites drawn uniformly without replacement from $V(x) \setminus \{v\}$, and let $\widetilde{L}$ be Equation (1) computed with $\widetilde{W}_v$. Then $\mathbb{E}[\widetilde{L}] = s + (1 - s)\, \frac{n-1}{N-1}$, and the mean of M independent draws, $L^{\text{ctrl}}(v)$, has the same expectation. This is Equation (2).*

Proof. Proposition 1 applies to $\widetilde{W}_v$, which contains v, so $\widetilde{L} = s + (1 - s)\widetilde{L}_{\text{off}}$. The quantities s and $\sum_{u \neq v} A(u)$ are fixed by the clean and perturbed runs and do not depend on the draw. Each $u \neq v$ enters $\widetilde{W}_v$ with probability $(n - 1)/(N - 1)$, so by linearity of expectation $\mathbb{E}[\widetilde{L}_{\text{off}}] = \frac{n-1}{N-1}$. The mean of M draws is a linear combination with weights summing to one. For $s = 1$ both sides equal 1. □

Corollary 1 (What the excess measures)**.** $L^{(r)}(v) - \mathbb{E}[L^{\text{ctrl}}(v)] = (1 - s)\big(L_{\text{off}} - \frac{n-1}{N-1}\big)$. *For $s < 1$ the expected excess is positive exactly when the off-diagonal locality exceeds $(n - 1)/(N - 1)$, the off-diagonal chance level of a relation of the same size, and its magnitude scales with the non-self mass $1 - s$.*

Proof. Subtract Proposition 2 from Proposition 1. □

Corollary 2 (Shared null)**.** *If the effect mass is uniform over admissible sites, $A(u) = c > 0$ for all u, then $L^{(r)}(v) = \mathbb{E}[L^{\text{ctrl}}(v)] = n/N$. Averaged over sources, both give the geometry-matched null $b_r(x)$ of Section 3.1.*

Proof. Uniform mass gives $s = 1/N$ and $L_{\text{off}} = (n - 1)/(N - 1)$. Substituting into either proposition gives $1/N + \big(1 - 1/N\big)(n - 1)/(N - 1) = n/N$. □

Corollary 3 (Source-excluded functionals)**.** *A functional that excludes the source from both sums, as the MTU3D functional does, equals L_{off}. Its size-matched control has expectation $(n - 1)/(N - 1)$, and averaged over sources this is the fraction of off-diagonal pairs inside the neighborhood, the null against which that functional is read. The control therefore adds nothing to the null there, as Section 4.4 states.*

Proof. With the source removed, the functional is L_{off} by definition, and the second step of the proof of Proposition 2 gives its expectation under $\widetilde{W}_v$. Within one example N is fixed, so the source average of $(n - 1)/(N - 1)$ is $\sum_v (n_r(v) - 1) / \sum_v (N - 1)$. □

Scope. These are identities per source and per edge. The reported scores average over sources and examples, and Table 2 compares paired differences between two channels whose self-mass shares differ, so Corollary 1 does not make the sign agreement between the excess and Δ^{off} in all fourteen rows of that table a consequence of the definitions. It explains why the agreement is expected and leaves it an empirical finding.

Table 4: Probe suite used to instantiate interaction locality. The main text prioritizes finite perturbations as causal evidence; SAE features provide semantic readability, and appendix-only structural probes report linearized topology and attention bias.

Probe	Measurement	What it supports / limitation
SAE features	Ablation impact within task segments	Human-readable feature locality; may miss distributed collective effects.
Finite patching	Distance decay under source perturbations	Causal reach at finite noise scale, with reliability checks.
Output and flip readouts	Logit-change and prediction-flip locality on the same edge	Connects measured spread to task behavior; the flip metric is not derivable from the state change.
Structural checks (appendix)	Jacobian and attention locality	Linearized or architectural topology, not the primary causal claim.

Table 5: Claim-level triangulation. The framework connects semantic readability, finite-intervention locality, and structural topology so each mechanistic claim has a dedicated source of evidence.

Mechanistic claim	Main evidence	Supporting detail
Step 1 is a meaningful analysis window	Hidden-state drops and step-1 decodes show early solution structure	Patching and structural kernels analyze the causal interactions inside this window.
SAE features provide readable segment hypotheses	Top features align with corridors and Sudoku boxes in representative cases	Representative feature examples are in Appendix F.1.
H-level writes tend to use the local same-segment channel	Activation patching gives a positive paired within-H minus within-L difference in all six HRM/TRM cells	The small HRM/Sudoku gap is consistent with a scale-selection effect, a Sudoku house already being a broad local unit; Appendix D.1 reports its sensitivity to calibration refitting.
Cross-cycle carry is broader than the same-phase write	Maze patching separates concentrated within-H writes from broader cross-cycle H-state effects	Raw drops are reported in Appendix D.5.
MTU3D carries causal spatial locality at a module boundary and throughout its encoder	Stage 1–2 input ablation is local over 200 scenes, all density strata, and four checkpoints, and locality is above baseline at every encoder depth	Checkpoint controls are in Appendix G.3.
Structural Jacobian topology contextualizes the main claim	Appendix kernels recover linearized cell/segment topology	These diagnostics show how the same task geometry appears in local linear interactions.

A.3 Probe suite and secondary probes

SAE semantic locality. For H and L activations at the critical cycle, we train independent sparse autoencoders ($512 \to 2048$, $\lambda_1 = 10^{-3}$), rank features by ablation impact, and compute how much each feature's effect remains within a task segment (corridor, Sudoku house, or ARC object). SAE locality is human-readable but feature-wise: it reveals where individual sparse features act, not necessarily the aggregate topology of the whole update.

A.4 Claim-level triangulation details

What the probes estimate. The probes share a geometry question but estimate different objects, so "triangulation" should be read as complementary testing rather than interchangeable confirmation: agreement strengthens a claim, and disagreement localizes which evidential level does not imply another.

Probe	Quantity estimated	Reading when it is near baseline
Jacobian	Infinitesimal differential coupling	A distributed, locally linear topology can still be present.
Finite patching	Realized nonlinear response to a calibrated intervention	Structural sensitivity need not become finite causal concentration at that intervention scale.
SAE ablation	Dependence on one learned sparse feature	The relation need not be isolated in any single sparse feature.

The Sudoku constraint breakdown of Appendix D.6 is the worked example: column-constraint structure is visible to the Jacobian while finite patching and SAE ablation stay near baseline. The output-level analysis of Section 4.3 shows separately that the finite H-channel effects we do observe reach logits and discrete predictions.

B Experimental configuration

B.1 Checkpoint provenance, sample size, and acceptance

Checkpoints. Every cell in Table 1 and the appendices loads one of the following. Released checkpoints are identified by repository; the two trained for this work are identified by step and are reproducible from the released code and configuration.

Cell	Task	Source	Step / revision
HRM	Maze 30×30 hard	`sapientinc/HRM-checkpoint-maze-30x30-hard`	released
HRM	Sudoku-Extreme	`sapientinc/HRM-checkpoint-sudoku-extreme`	released
HRM	ARC-AGI-2	`sapientinc/HRM-checkpoint-ARC-2`	released
TRM	Maze 30×30 hard	trained here (see below)	`step_195310`
TRM	Sudoku-Extreme	`Sanjin2024/TRM-Sudoku-Extreme-att`	`step_21700`
TRM	ARC-AGI-1	`Sanjin2024/TRM-ARC-AGI-1`	`step_155718`
Non-recursive	Maze 30×30 hard	trained here, same codebase	`step_156240`
MTU3D	ScanNet	`bigai/MTU3D`, `stage2-pretrain-all`	released

Repository names are abbreviated: the two `Sanjin2024/TRM-...` entries are `TinyRecursiveModels-Sudoku-Extreme-att` and `TinyRecursiveModels-ARC-AGI-1`. The MTU3D conditions table additionally loads the three released fine-tunes `stage2-fine-tune-{ovon,goat,sg3d}` (Appendix G.3). The non-recursive baseline is `transformers_baseline@Model_ACTV2` with eight layers, `H_cycles`=1 and `halt_max_steps`=16; forty steps were saved and eight evaluated (Appendix E.1). The fine-tuning runs of Section 5 all warm-start from TRM/Maze `step_195310` and are evaluated at `step_468`. ScanNet features are user-supplied and not redistributable.

Why TRM/Maze is trained rather than downloaded. It is the one cell whose checkpoint we produced. When these experiments began, to our knowledge, the only public TRM/Maze checkpoint was undertrained — roughly 9k steps at 78.7% path-cell accuracy, which under-reproduces the reported behaviour of the architecture and cannot support a locality measurement. We therefore trained the model from the released training script and a committed configuration, saving on a geometric schedule to `step_195310`, so that the cell rests on a checkpoint reproducible from this repository rather than on an artifact we could not regenerate.

A competitively trained public checkpoint, `maze-trm-yagiz`, became available afterwards. We did not substitute it; we ran it through the identical $n = 500$ pipeline as a compatibility check and report it as its own row:

TRM/Maze checkpoint	within-L	within-H	Δ [95% CI]
`step_195310` (used throughout)	.032 [.032, .032]	.263 [.261, .265]	+.230 [+.228, +.232]
`maze-trm-yagiz` (public)	.031 [.031, .031]	.426 [.422, .430]	+.395 [+.392, +.399]

Both are behaviorally competitive — path-cell accuracy .984 and .983, clean-correct 100% — and the public checkpoint is in fact higher on exact match (.835 against .725). This is therefore not an undertrained-versus-trained comparison; that description applies only to the 9k checkpoint we

rejected. The two agree on direction and differ in magnitude, so within-H magnitude is checkpoint-dependent while within-L and the sign of Δ are checkpoint-robust. Note which way the choice cuts: standardizing on `step_195310` lowers the reported within-H from .426 to .263 and Δ from $+.395$ to $+.230$, so it costs effect size and buys reproducibility.

Sample size required to resolve each effect. For each cell we report the mean and standard deviation of the per-example paired difference, the smallest n whose paired CI excludes zero, and the n required for a $\pm.01$ CI half-width. Only the two ARC cells have per-example variance large enough to require substantial sampling; $n = 500$ clears both.

Architecture	Task	mean Δ	sd	n (CI excl. 0)	n (CI $\pm.01$)
HRM	Maze	.078	.039	1	42
HRM	Sudoku	.043	.012	1	4
HRM	ARC-AGI	.100	.094	3	239
TRM	Maze	.230	.024	1	16
TRM	Sudoku	.100	.010	1	3
TRM	ARC-AGI	.061	.069	4	129

Acceptance and effective sample size. The $n = 500$ examples per cell are accepted examples, and no accepted example was removed for zero total effect (0 of 500 in all six cells). For ARC, the 2–15-component eligibility rule defines the population from which analysis examples are drawn, so n is a post-filter count rather than a nominal count later reduced by exclusions. Each accepted example contributes all valid source and target sites; these within-example measurements are not treated as independent samples, and confidence intervals resample examples, making the effective inferential n equal to the number of examples.

B.2 Measurement configuration

Per-cell choices. Each model–task combination fixes its edges, kernel, calibration, and sample size before model responses are inspected.

Model / task	Edges measured	Kernel W_r	Calibration	n
HRM, TRM / Maze	within-L, within-H, cross-HH, input→output	path distance ≤ 1	self-drop .30	500
HRM, TRM / Sudoku	within-L, within-H, cross-HH	3×3 box; row, column	self-drop .30	500
HRM, TRM / ARC-AGI	within-L, within-H, cross-HH	connected component; color, radius	SNR $= 1$	500
Non-recursive / Maze	input→output	path distance ≤ 1	self-drop .30, per model	300
MTU3D / ScanNet	object feature → encoder layers; layer-1 slot → encoder layers	per-scene radius ($\approx$7 neighbors)	zero ablation; scaled perturbation	200 scenes

The non-recursive baseline is `baseline_maze_ext` at `step_156240` (24,499 cumulative epochs), reaching 96.2% path-cell accuracy against TRM's 98.0%, and its comparison uses $n = 300$ common-correct mazes; accuracies and the full training curve are reported with the comparison in Section 4.3. Calibration is behavioral rather than absolute, so it is set separately for each model. Across the recursion axis this means the two models are compared at equal behavioral perturbation strength, and we therefore report the comparison at several calibration levels (Section 4.3).

B.3 Experimental details

MTU3D/ScanNet setting. For the 3D embodied extension, MTU3D is the model and ScanNet is the task source: we analyze precomputed object-level Stage-1 features from 200 ScanNet indoor scenes (Dai et al., 2017). The measured task is scene grounding over 3D object/voxel representations. Locality neighborhoods are defined by metric object distance, with a per-scene radius

calibrated so that objects have about seven neighbors inside it on average, which keeps locality comparable across scenes of different object density; the random baseline is the corresponding fraction of near-object off-diagonal pairs. The MTU3D experiments therefore test whether input corruption changes spatially nearby scene objects more than distant objects.

Compute resources. All reported analyses use released checkpoints and do not require retraining the HRM/TRM models. The accompanying code is written in PyTorch and is intended to run on a single CUDA GPU; the repository recommends a GPU with at least 6GB VRAM for notebook-level analyses, with selective state capture and batch size 1 used for memory-intensive activation and intervention runs. The setup script downloads dependencies, datasets, and checkpoints; checkpoint sizes range from tens of MB for small TRM checkpoints to approximately 2GB for ARC-AGI checkpoints. The finite-noise activation-patching, SAE-ablation, and Jacobian diagnostic scripts process examples independently, so runtime scales approximately linearly with the number of evaluated samples. Reproducing the optional Maze-TRM training run from scratch is documented as requiring roughly 40 hours on a single A100 40GB GPU, but the paper's main experiments use the released/analyzed checkpoints rather than retraining them.

B.4 Licences

Licence. The official HRM implementation is released under Apache-2.0 and links to the official Sapient checkpoint artifacts used in this work. The checkpoint pages are publicly released by the same authors, but do not expose separate license metadata beyond the official repository release context.

B.5 Locality-aware fine-tuning: configuration and training dynamics

Table 3 shows that the measured quantity is steerable, with three properties that are hard to explain as an artifact of any single contrast. Within-H locality is *monotone* in the regularizer strength, rising from .272 at the $\beta = 0$ control through .279 at $\beta = 0.01$ to .308 at $\beta = 0.1$. The seed ranges of the control and the strongest arm are *disjoint*, $[.270, .275]$ against $[.305, .313]$, and the gap between them ($+.037$) is roughly five times the within-arm spread. And cell accuracy is preserved throughout (.9800–.9833), so the gain is not bought with task performance. Within-L is unchanged at .032 in every arm, which is what the objective predicts: it penalizes off-neighborhood mass in the H-write only.

The effect is modest — about 13% over the matched control at $\beta = 0.1$ — and measured on one task with one architecture. What it establishes is that the score is differentiable, that optimizing it moves the quantity it is defined to measure, and that it does so without degrading accuracy.

What the gain is made of. Within-L is unchanged, but the objective touches only the write, so that constancy cannot say whether the gain is propagation to neighboring sites or retention at the perturbed site. Table 6 splits the within-H score of every arm by Proposition 1 into the self-mass share s and the off-diagonal locality L_{off}; the identity holds per sample to 7×10^{-8}. Both parts rise. From the released checkpoint to $\beta = 0.1$, s rises by $+.0448$ $[+.0409, +.0488]$ and L_{off} by $+.0031$ $[+.0021, +.0041]$, and against the $\beta = 0$ control by $+.0356$ $[+.0323, +.0388]$ and $+.0035$ $[+.0028, +.0042]$; the seed ranges of $\beta = 0$ and $\beta = 0.1$ are disjoint for both. The self-mass change accounts for 95% of the raw rise, so the gain is mostly retention, and off-diagonal locality, the quantity conditional on the mass having left the source, shows a smaller real gain in propagation. The excess over the size-matched control is nearly flat, $+.0002$ $[-.0004, +.0009]$ against the released checkpoint and $+.0010$ $[+.0005, +.0014]$ against $\beta = 0$. By Corollary 1 the excess is $(1-s)(L_{\text{off}} - r)$, and the rising s shrinks the factor $1 - s$ from .786 to .742, which cancels most of the off-diagonal gain. Intervals are percentile bootstraps over samples (2000 draws, seed 42). Two caveats apply. The raw column is a re-measurement at the same shared σ and agrees with Table 3 to within .0012 in every arm. The arms are compared unpaired because the maze identities were not recorded, so pairing would only tighten these intervals.

How the measurement is calibrated matters here more than elsewhere. Locality is defined at a calibrated perturbation scale, and our default is to calibrate σ per model so that different models are perturbed at equal behavioral strength. That convention does not suit this experiment: the auxiliary

Table 6: Within-H locality of the fine-tuning arms, split into retention at the perturbed site and propagation to other sites. Maze-TRM, $n = 200$, read at the shared evaluation scale $\sigma = 12.508$ used in Table 3. s is the self-mass share, L_{off} the locality with the source removed from both sums, and the control $s + (1 - s)r$ follows from Equation (2); $r = (n_r(v) - 1)/(|V(x)| - 1)$ is the off-diagonal chance level of Proposition 2. The split is exact: raw $= s + (1 - s)L_{\text{off}}$ per sample. Bracketed ranges are the minimum and maximum over three seeds, not bootstrap intervals; the released checkpoint is a single model.

Variant	within-H (raw)	self s	L_{off}	control	excess
Released	0.2628	0.2137	0.0625	0.2274	0.0353
$\beta = 0$ (control)	0.2711	0.2229	0.0621	0.2365	0.0346
	[0.2689, 0.2740]	[0.2209, 0.2254]	[0.0617, 0.0628]	[0.2345, 0.2389]	[0.0343, 0.0350]
$\beta = 0.01$	0.2780	0.2298	0.0627	0.2433	0.0347
	[0.2769, 0.2801]	[0.2288, 0.2317]	[0.0625, 0.0631]	[0.2423, 0.2452]	[0.0346, 0.0349]
$\beta = 0.1$	0.3070	0.2585	0.0656	0.2715	0.0356
	[0.3029, 0.3117]	[0.2550, 0.2630]	[0.0644, 0.0663]	[0.2681, 0.2759]	[0.0349, 0.0359]

loss changes how sensitive the model is to a perturbation of a given size, so per-checkpoint recalibration lets a change in sensitivity move the score independently of any change in propagation. Across these checkpoints the calibrated σ spans 9.4–15.6 and its rank correlation with within-H locality is -0.94. At a shared σ, which is also how the auxiliary loss was applied during training, the seed ranges are tight and the arms are ordered; at each model's own σ they overlap, the ranges widen roughly thirtyfold, and $\beta = 0.01$ scores above $\beta = 0.1$. We report both columns in Table 3, and the comparison yields a reusable rule: an intervention that alters noise sensitivity should be evaluated at a fixed perturbation scale.

Configuration. Nine fine-tuning runs: $\beta \in \{0, 0.01, 0.1\}$ crossed with seeds $\{0, 1, 2\}$, each warm-started from the released Maze-TRM checkpoint and trained for 60 epochs with the auxiliary loss applied to the final gradient-carrying H-write at a fixed auxiliary scale $\sigma = 9.484$. Evaluation uses $n = 200$ mazes. Scope is Maze only; the Sudoku figures quoted elsewhere are single-run.

Why the primary measurement holds σ fixed. Interaction locality is defined at a calibrated perturbation scale, and for cross-model comparison we normally calibrate σ per model so that each is perturbed at equal behavioral strength. That convention is the wrong one here. The auxiliary loss changes how sensitive the model is to a perturbation of a given size, so recalibrating per checkpoint lets a change in sensitivity move the score independently of any change in propagation: across the fine-tuned checkpoints the calibrated σ ranges from 9.4 to 15.6, and its rank correlation with within-H locality is -0.94. Holding σ at the released checkpoint's value, which is also how the auxiliary loss itself was applied during training, separates the two and shrinks the across-seed spread by roughly a factor of thirty. We report both conventions in Table 3 and treat their disagreement as part of the result: it shows that a training intervention which alters noise sensitivity cannot be evaluated with a sensitivity-dependent calibration.

Interval semantics. Bracketed ranges are the minimum and maximum across the three seeds, not bootstrap intervals, because the quantity of interest is variation between training runs. Within-run bootstrap intervals are far tighter — the released checkpoint's within-H interval is $[.261, .267]$ at $n = 200$ — so quoting them alone would overstate the precision of any single arm.

The ordering survives every run. Across all ten checkpoints evaluated here, within-H exceeds both within-L and cross-HH in every run. Within-L is unchanged by the intervention ($.032$ in every arm), which is what the objective predicts: it penalizes off-neighborhood mass in the H-write only.

C Convergence and anchors

C.1 Convergence and the choice of anchor

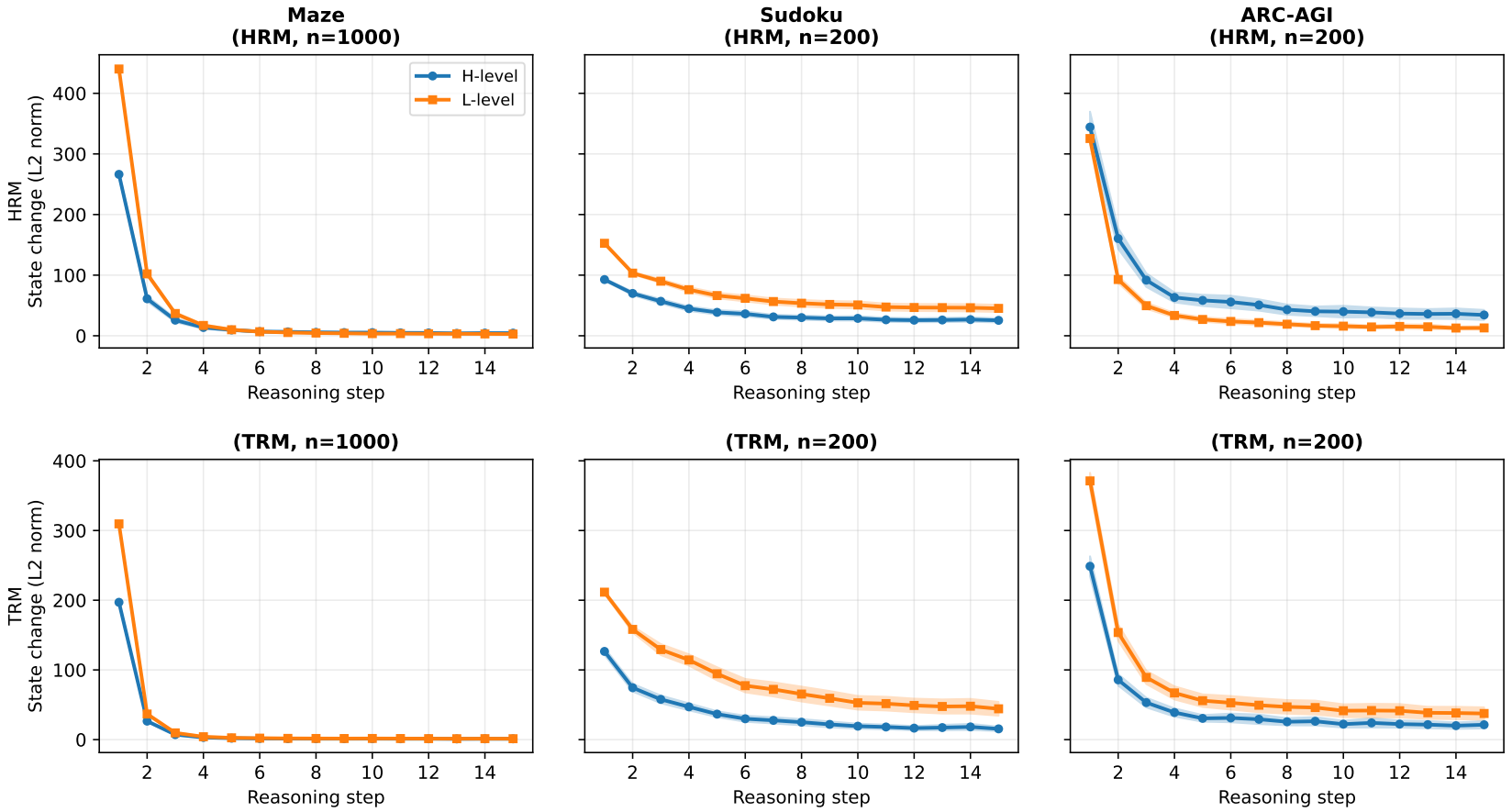


Figure 4: H/L hidden-state change across reasoning steps for HRM (top) and TRM (bottom). Blue tracks high-level changes and orange low-level; bands are 95% bootstrap CIs over examples ($n = 1000$ Maze, $n = 200$ Sudoku and ARC-AGI). The three tasks converge differently, which is why the analysis window is selected per task.

The anchors are the phase and call at which the plotted change has plateaued: HRM phase 2, call 1 on every task; TRM phase 2, call 3 on Maze and ARC-AGI and call 5 on Sudoku.

C.2 Additional convergence results

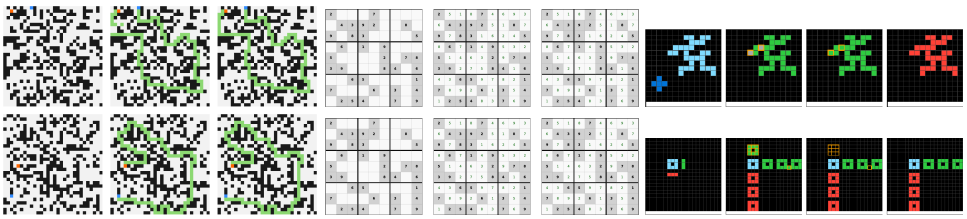


Figure 5: Initial, step-1, and final decoded outputs for all model–task pairs (HRM top row, TRM bottom row; columns are Maze, Sudoku, ARC-AGI). Each panel shows the decoded task structure at the input/initial state, after the first reasoning step, and at the final step; colors follow the task-specific output convention for paths, digits, or ARC objects. Step 1 already contains recognizable task structure across domains, supporting it as a solution-forming window.

Cycle-level diagnostics support the critical-cycle choices reported in Section 3.3.

C.3 Additional first-step cycle decodes

Figures 7 to 10 add first-step cycle decodes to those of Figure 5.

D Robustness of the ordering

D.1 Robustness of the locality ordering

Table 7: Marginal locality of each channel with its geometry-matched baseline, $n = 500$. within-L is one low-level call, within-H the same-phase final $L \to H$ write, and cross-HH the next-phase $H \to H$ carry. Cross-cycle carry is less concentrated than the same-phase write in every cell, and both within-cycle channels exceed the baseline everywhere. Percentile-bootstrap 95% CIs over examples.

Architecture	Task	Baseline	within-L	within-H	cross-HH
HRM	Maze	.026	.266 [.265, .267]	.344 [.341, .347]	.220 [.218, .223]
HRM	Sudoku	.111	.332 [.331, .332]	.374 [.373, .375]	.339 [.338, .339]
HRM	ARC-AGI	.367	.477 [.465, .489]	.577 [.563, .590]	.526 [.512, .539]
TRM	Maze	.026	.032 [.032, .032]	.263 [.261, .265]	.060 [.059, .060]
TRM	Sudoku	.111	.113 [.113, .113]	.213 [.212, .214]	.204 [.202, .205]
TRM	ARC-AGI	.379	.627 [.616, .638]	.688 [.675, .702]	.457 [.443, .472]

Each entry is the paired difference $\Delta = L_{\text{within-H}} - L_{\text{within-L}}$ at the stated setting, $n = 500$ throughout (one setting $n = 499$). Underlined entries reverse the ordering; * marks the calibration used for the main results in each cell.

Noise distribution and direction. Variance-matched distribution families and magnitude-matched direction families.

Setting	HRM/Maze	HRM/Sudoku	HRM/ARC	TRM/Maze	TRM/Sudoku	TRM/ARC
uniform	+0.112	−0.017	+0.090	+0.246	+0.120	+0.081
Rademacher	+0.113	−0.017	+0.087	+0.249	+0.120	+0.077
Laplace	+0.113	−0.017	+0.087	+0.245	+0.120	+0.079
random unit	+0.114	−0.017	+0.089	+0.249	+0.120	+0.077
top PC	+0.008	+0.024	+0.075	+0.018	+0.020	+0.082
bottom PC	+0.107	−0.026	+0.098	+0.298	+0.354	+0.075

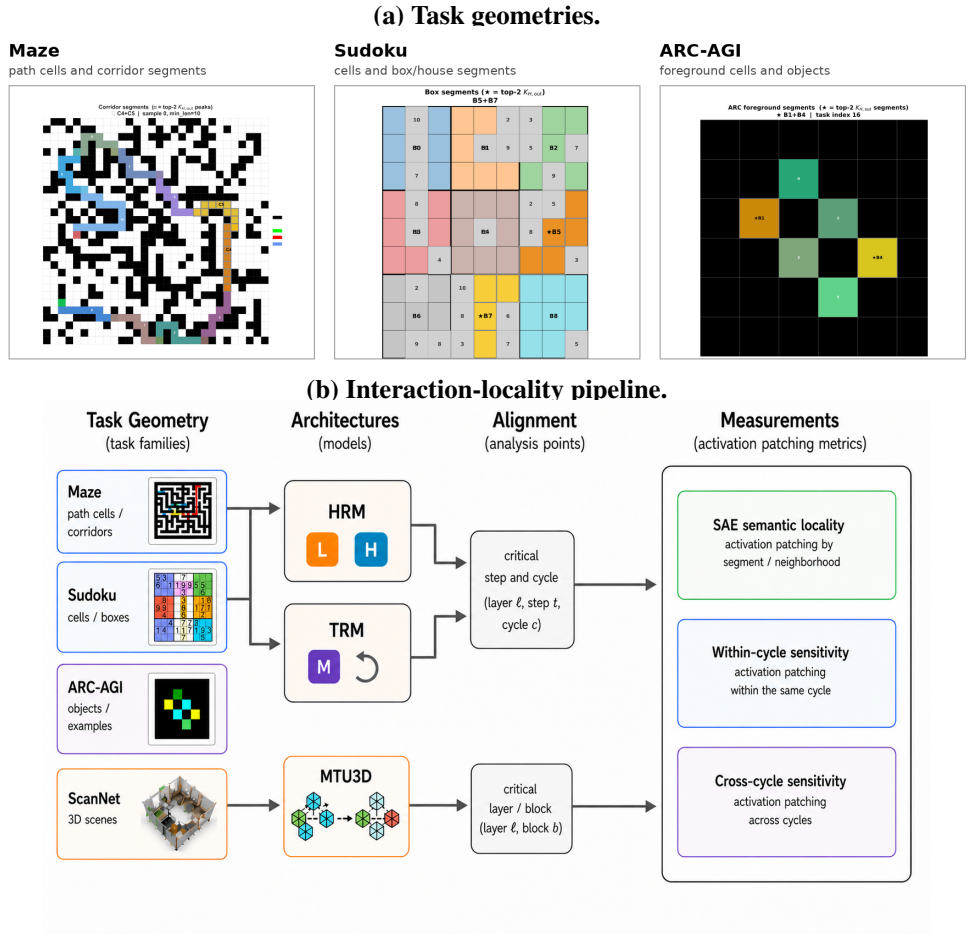


Figure 6: Interaction-locality framework. Panel (a) shows the task geometries that define locality: Maze path cells and corridors, Sudoku cells and 3×3 boxes, and ARC-AGI foreground objects/examples. Panel (b) shows how the same measurement suite is aligned across models: HRM and TRM are analyzed at a critical recursive step and cycle, while MTU3D/ScanNet is analyzed at critical layers or blocks. The green measurement box denotes SAE semantic locality by segment/neighborhood, the blue box denotes within-cycle or within-layer finite-noise patching sensitivity, and the purple box denotes cross-cycle or cross-layer patching sensitivity. Structural Jacobian and attention diagnostics are reported in the appendix as topology checks rather than the primary finite-intervention evidence.

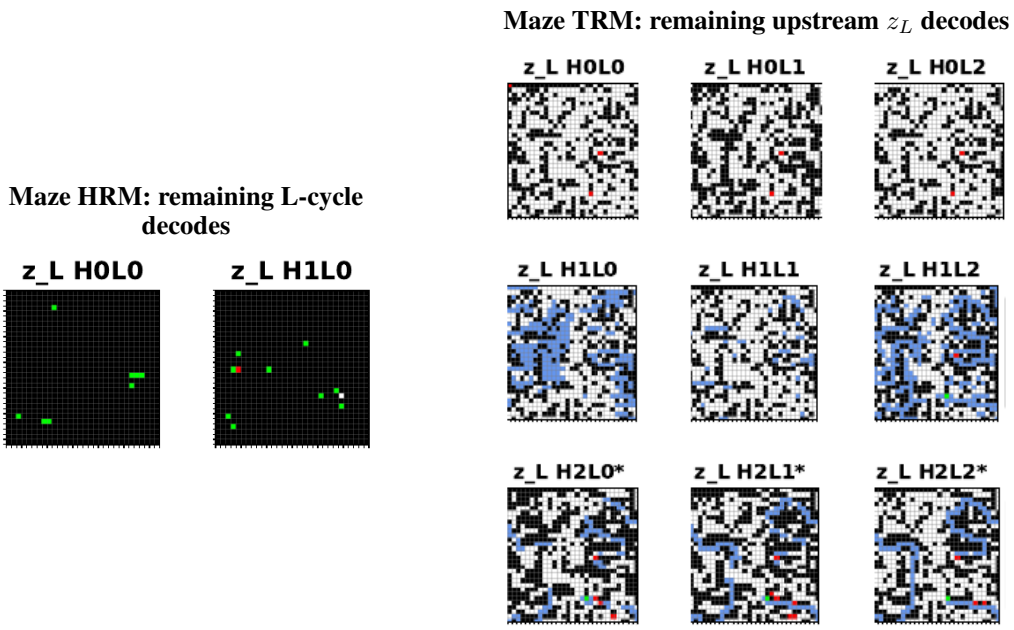


Figure 7: Maze cycle-decode panels not shown in the main selected figure. These additional observational decodes restore the first-step cycle context while keeping the main text focused on H-update endpoints.

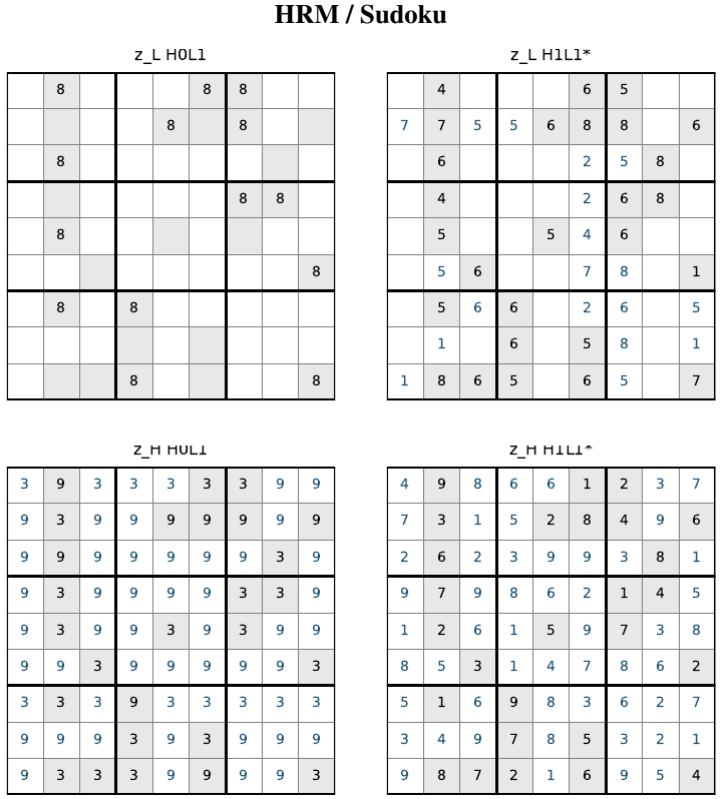


Figure 8: Selected first-step cycle decodes for Sudoku HRM. The two columns show the H-cycle endpoints H0L1 and H1L1, with z_L on top and z_H below. The decoded grids are observational sanity checks rather than causal evidence.

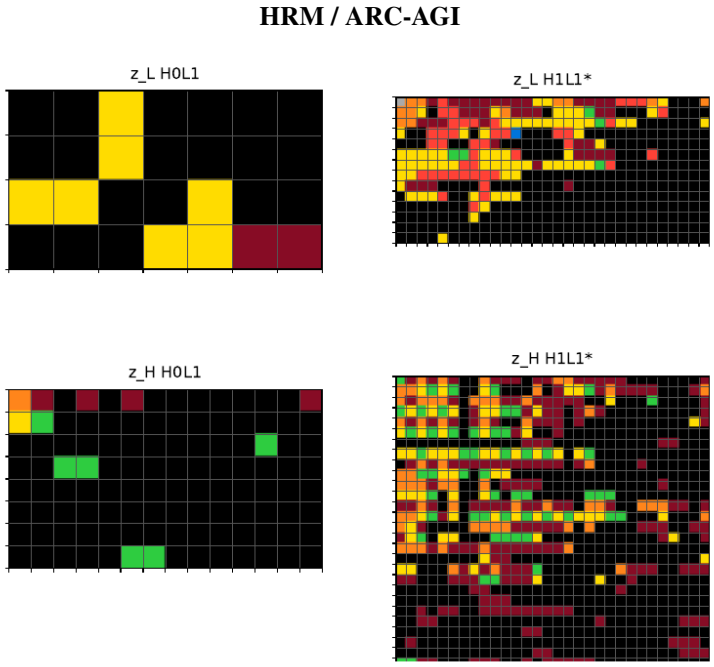


Figure 9: Selected first-step cycle decodes for ARC-AGI HRM. The two columns show H-cycle endpoints, with z_L on top and z_H below.

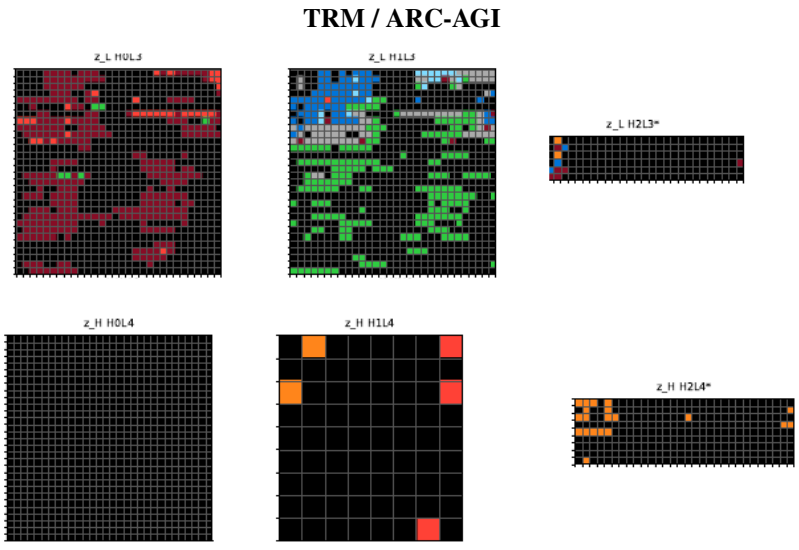


Figure 10: Selected first-step cycle decodes for ARC-AGI TRM. As in Maze and Sudoku, these panels show what is decodable at each cycle, while the Jacobian analysis measures which positions influence later updates.

The ordering is preserved in 31 of 36 checks. Every failure is the same cell: HRM/Sudoku reverses under five of the six draws, while the other five model–task cells preserve the ordering in all 30 checks.

Perturbation scale. Self-drop calibration targets a given rate of clean-correct prediction change at the perturbed site; SNR calibration matches perturbation magnitude to activation magnitude.

Setting	HRM/Maze	HRM/Sudoku	HRM/ARC	TRM/Maze	TRM/Sudoku	TRM/ARC
self-drop 10%	$+0.022$	-0.023	$+0.110$	$+0.451$	$+0.212$	$+0.049$
self-drop 30%	$+0.114^{*}$	-0.017^{*}	$+0.089$	$+0.249^{*}$	$+0.120^{*}$	$+0.077$
self-drop 50%	$+0.133$	$+0.052$	$+0.136$	$+0.070$	$+0.040$	$+0.066$
SNR 0.5	-0.048	$+0.016$	$+0.072$	-0.031	-0.006	$+0.028$
SNR 1	-0.057	-0.011	$+0.101^{*}$	-0.030	$+0.003$	$+0.055^{*}$
SNR 2	-0.035	-0.002	$+0.116$	$+0.020$	$+0.034$	$+0.083$

The ordering is preserved in 26 of 36 checks, and the failures separate cleanly by calibration family. Under behaviorally calibrated self-drop the ordering holds in 15 of the 15 checks outside HRM/Sudoku; under matched SNR it holds in only 9 of 15. Matched-SNR calibration reverses HRM/Maze at all three levels and TRM/Maze at two, which is the sense in which the result is not scale-free: fixing perturbation magnitude relative to activation magnitude is not the same intervention as fixing it relative to behavior.

HRM/Sudoku is the unstable cell. Across the twelve settings HRM/Sudoku ranges from $-.026$ to $+.052$ and changes sign seven times, and it is the only cell whose reference-setting sign disagrees between runs: the sweep's reference calibration gives $\Delta = -.017$ where Table 1 gives $+.043$ and the kernel run of this appendix gives $+.119$, all at the same nominal target and all at $n = 500$. Sample size is therefore excluded as the cause. The disagreement traces to the within-L scale. Within-H uses $\sigma = 3.134$ in both runs; within-L uses 12.630 in the main table and 5.428 in the sweep. The main-table value overshoots its target, with an achieved self-drop of $.615$ $[.550, .680]$ against $.30$ (Table 29), and an independent refit gives 6.062, close to the sweep's. The larger scale spreads within-L's effect and lowers its locality from $.401$ to $.332$, and that drop produces the main table's positive Δ. The two fits diverge because per-analog σ is fitted by a stochastic search whose noise draws are not seeded independently of run order. This cell has the smallest effect in the suite, so its sign is not stable across calibrations, and we do not rest any claim on it. The five remaining cells reproduce the main-table ordering.

Resample patching. Table 8 replaces the source site's state with the same site's state from the next accepted example, so the substituted vector is one the model produces there and no scale is chosen. The raw ordering holds in all six cells. The achieved self-drop is $.089$ to $.611$, observed and not targeted, against $.005$ to $.050$ under matched SNR, so the intervention is both on-distribution at the perturbed site and behaviorally consequential. Downstream it is not: the capture-site ratio of Appendix H.1 is below one in four cells, and 3.14 and 2.55 in the two Maze cells, where the substitution moves the write beyond the natural spread. With the source excluded, the difference reverses in HRM/Maze as under noise, and also in TRM/Maze and TRM/ARC-AGI, so the cells in which the write's concentration reaches other sites depend on the intervention family. HRM/Sudoku keeps a positive raw ordering at $+.006$, the smallest effect in the table, where both calibrated families reverse it; its sign depends on the intervention family and we rest no claim on it. On ARC-AGI the admissible sites are the foreground, which differs per example, so sources and targets are restricted to sites admissible in both donor and recipient, and pairs sharing fewer than eight are dropped, leaving 177 and 134 of 500. The median pair shares three or four sites, so the retained pairs lean toward larger foregrounds and the ARC-AGI rows describe that subset; the Maze and Sudoku rows use all 500 examples.

Relation kernels. Alternative predeclared kernels at $n = 500$, holding the model, edge, endpoint, and calibration target fixed and changing only W_r.

Table 8: Resample patching: the source site's state is replaced by the same site's state from a different accepted example, so no perturbation scale is chosen. Δ and Δ^{off} are paired per example with percentile-bootstrap 95% CIs over examples; ARC-AGI keeps only donor–recipient pairs sharing at least eight admissible sites (Appendix D.1). Self-drop is observed, not targeted, and the capture ratio is the downstream displacement of Appendix H.1 over the natural spread. The last two columns repeat Δ at self-drop 30% and SNR 1 from the perturbation-scale check of Appendix D.1.

Cell	n	Δ	Δ^{off}	self-drop	cap. ratio	$\Delta_{\text{self-drop}}$	Δ_{SNR}
HRM/Maze	500	$+.1332$ $[+.1301, +.1365]$	$-.0306$ $[-.0315, -.0297]$	.611	3.14	$+.1136$	$-.0572$
TRM/Maze	500	$+.1603$ $[+.1572, +.1633]$	$-.0368$ $[-.0379, -.0356]$	.279	2.55	$+.2486$	$-.0297$
HRM/Sudoku	500	$+.0060$ $[+.0052, +.0067]$	$+.0579$ $[+.0572, +.0585]$	.589	0.70	$-.0172$	$-.0106$
TRM/Sudoku	500	$+.0223$ $[+.0206, +.0240]$	$+.0162$ $[+.0146, +.0178]$	.089	0.39	$+.1201$	$+.0030$
HRM/ARC-AGI	177	$+.1039$ $[+.0887, +.1210]$	$+.0711$ $[+.0582, +.0842]$	.275	0.32	$+.0892$	$+.1007$
TRM/ARC-AGI	134	$+.0262$ $[+.0162, +.0371]$	$-.0197$ $[-.0305, -.0074]$	.233	0.44	$+.0767$	$+.0554$

Model / task	Kernel	Baseline	within-L	within-H	Δ [95% CI]
HRM / Sudoku	box	0.111	0.265	0.384	$+0.119$ $[+0.118, +0.120]$
	row	0.111	0.307	0.384	$+0.078$ $[+0.077, +0.079]$
	column	0.111	0.260	0.297	$+0.037$ $[+0.036, +0.037]$
TRM / Sudoku	box	0.111	0.113	0.248	$+0.135$ $[+0.134, +0.136]$
	row	0.111	0.113	0.238	$+0.125$ $[+0.124, +0.126]$
	column	0.111	0.112	0.212	$+0.100$ $[+0.099, +0.101]$
HRM / ARC-AGI	component	0.367	0.477	0.577	$+0.100$ $[+0.092, +0.108]$
	color	0.591	0.653	0.723	$+0.070$ $[+0.064, +0.077]$
	radius	0.247	0.397	0.491	$+0.094$ $[+0.086, +0.101]$
TRM / ARC-AGI	component	0.379	0.627	0.688	$+0.061$ $[+0.055, +0.067]$
	color	0.638	0.788	0.826	$+0.038$ $[+0.033, +0.044]$
	radius	0.322	0.581	0.635	$+0.054$ $[+0.048, +0.060]$

Every tested kernel preserves a positive paired difference, including the HRM/Sudoku column kernel, the smallest of the twelve.

Outer step and recurrence index. Each entry is the paired difference at the stated endpoint, $n = 500$ throughout. Conditions C_k read the ordering at the kth h-cycle endpoint; conditions S_t read it after outer reasoning step t. TRM has a third h-cycle; HRM does not.

Endpoint	HRM/Maze	HRM/Sudoku	HRM/ARC	TRM/Maze	TRM/Sudoku	TRM/ARC
cyc h_0	$\underline{-0.055}$	$+0.143$	$+0.081$	$+0.285$	$+0.031$	$+0.194$
cyc h_1	$\overline{+0.078}$	$+0.118$	$+0.101$	$+0.176$	$+0.051$	$+0.122$
cyc h_2	—	—	—	$+0.231$	$+0.075$	$+0.055$
step 1	$+0.072$	$+0.162$	$+0.111$	$+0.263$	$+0.081$	$+0.085$
step 2	$+0.085$	$+0.166$	$+0.112$	$+0.264$	$+0.086$	$+0.087$
step 8	$+0.089$	$+0.182$	$+0.115$	$+0.264$	$+0.090$	$+0.087$
step 15	$+0.090$	$+0.182$	$+0.114$	$+0.264$	$+0.091$	$+0.087$

The ordering is preserved in 38 of 39 checks at $n = 500$. Every reasoning-step condition passes in every cell; the single exception is an internal recurrence endpoint, HRM at its first h-cycle ($\Delta = -.055$). The claim is therefore a later-cycle result: the ordering holds from the second h-cycle onward and at every outer reasoning step, but not at HRM's first h-cycle.

Table 9: Paired within-H minus within-L locality at each readout, computed per example and bootstrapped, so the interval uses the between-channel covariance that marginal intervals discard. All eighteen intervals exclude zero, and the separation is larger at the output than in the hidden state in five of six cells. Flip locality is conditional on both channels flipping at least one target; its paired n is given. The state column uses this run's own calibration refit and differs from Table 1 in three cells. Percentile bootstrap, 2000 draws, seed 42.

Arch.	Task	Δ state	Δ output	Δ flip	flip n
HRM	Maze	$+.0777$ [$+.0745$, $+.0810$]	$+.2399$ [$+.2366$, $+.2431$]	$+.2916$ [$+.2756$, $+.3076$]	500
HRM	Sudoku	$+.1177$ [$+.1167$, $+.1186$]	$+.1083$ [$+.1073$, $+.1092$]	$+.0366$ [$+.0340$, $+.0392$]	500
HRM	ARC-AGI	$+.1007$ [$+.0928$, $+.1092$]	$+.1646$ [$+.1555$, $+.1744$]	$+.1116$ [$+.0698$, $+.1514$]	174
TRM	Maze	$+.2306$ [$+.2284$, $+.2329$]	$+.2379$ [$+.2354$, $+.2405$]	$+.8089$ [$+.7927$, $+.8252$]	500
TRM	Sudoku	$+.0753$ [$+.0746$, $+.0760$]	$+.0964$ [$+.0955$, $+.0975$]	$+.0966$ [$+.0910$, $+.1029$]	500
TRM	ARC-AGI	$+.0554$ [$+.0500$, $+.0608$]	$+.1318$ [$+.1242$, $+.1401$]	$+.1103$ [$+.0668$, $+.1535$]	167

D.2 THE PERMUTATION FLOOR

The same rule as Equation (2) orders the Sudoku types: box, row and column share a control of $.031$ for HRM and $.068$ for TRM and differ only in excess, with the box largest in both models and the column smallest.

The one-sided permutation value in Table 2 is pinned at its design floor $1/(M+1)$ for any row more than about five control standard deviations from its control, so it cannot distinguish a comfortable row from a marginal one. Of the fourteen type-cell rows, thirteen are at least 7.7 standard deviations from their control and eleven are beyond 24, and the only row within five is the Sudoku column type for HRM at 3.4. We re-scored both Sudoku cells with $M = 1000$ control draws: that row moves from $p = .0198$ to $p = .0030$ and remains above its control, and the five other Sudoku rows drop to the new floor of $1/1001$ with excesses unchanged to within $.0002$. Maze and ARC-AGI are reported at $M = 100$, since every row there is beyond 7.7 standard deviations and a lower floor would change no verdict. The $|\mathrm{sd}|$ column carries the comparison that p cannot.

D.3 OUTPUT-LEVEL READOUTS

Table 9 repeats the paired within-H minus within-L comparison at the logit change and at the prediction flip, and Table 10 gives each channel's output-effect locality, flip locality and unconditional flip rate.

D.4 ARC-AGI VERSION CONTROL AND OUTPUT-LEVEL INTERVALS

ARC-AGI-1 versus ARC-AGI-2. HRM is released for ARC-AGI-2 and TRM for ARC-AGI-1, so the cross-architecture ARC comparison in Table 1 mixes benchmark versions. We therefore ran each model on the other version as well, at the same $n = 500$ used for the main results. All four rows preserve a positive paired within-H minus within-L difference with a 95% CI excluding zero, and the paired difference is similar across versions ($+.101$ and $+.088$ for HRM, $+.058$ and $+.051$ for TRM), so the version asymmetry does not drive the reported ordering. Each version carries its own geometry-matched baseline because the object-neighborhood sizes differ.

Condition	within-L	within-H	cross-HH	Base	Δ
HRM / ARC-AGI-2 (native)	0.477 [0.465, 0.490]	0.579 [0.565, 0.592]	0.527 [0.514, 0.541]	0.367	$+0.101$
HRM / ARC-AGI-1 (swap)	0.500 [0.489, 0.512]	0.588 [0.575, 0.602]	0.533 [0.520, 0.548]	0.378	$+0.088$
TRM / ARC-AGI-1 (native)	0.631 [0.619, 0.642]	0.689 [0.676, 0.703]	0.460 [0.445, 0.475]	0.379	$+0.058$
TRM / ARC-AGI-2 (swap)	0.606 [0.595, 0.619]	0.658 [0.644, 0.672]	0.444 [0.430, 0.459]	0.369	$+0.051$

The swap is non-degenerate in both directions: HRM solves 332 ARC-AGI-1 examples and TRM solves 256 ARC-AGI-2 examples, so the target n is reachable in every cell.

Table 10: The same single-source intervention read at three levels, $n = 500$ per cell, holding the valid targets and the geometry-matched baseline fixed. State locality uses the ℓ_2 activation change, output-effect locality the ℓ_2 logit change, and flip locality the discrete prediction change of Section 3.2, conditioned on at least one flipped target. *Rate* is the unconditional fraction of eligible targets whose prediction changes. All eighteen channel-by-cell flip localities exceed their geometry-matched baseline. State locality is computed under this run's own calibration refit and differs slightly from Table 1.

			State locality			Output-effect locality		
Arch.	Task	Base	L	H	X-HH	L	H	X-HH
HRM	Maze	0.026	0.265	0.343	0.220	0.132	0.372	0.218
HRM	Sudoku	0.111	0.266	0.384	0.339	0.259	0.367	0.325
HRM	ARC-AGI	0.367	0.477	0.578	0.527	0.395	0.560	0.508
TRM	Maze	0.026	0.032	0.263	0.060	0.028	0.266	0.057
TRM	Sudoku	0.111	0.119	0.194	0.209	0.115	0.211	0.205
TRM	ARC-AGI	0.379	0.631	0.687	0.455	0.524	0.655	0.441

		Flip locality			Flip rate		
Arch.	Task	L	H	X-HH	L	H	X-HH
HRM	Maze	0.278	0.570	0.360	0.016	0.007	0.014
HRM	Sudoku	0.240	0.276	0.263	0.098	0.054	0.076
HRM	ARC-AGI	0.442	0.674	0.617	0.010	0.016	0.026
TRM	Maze	0.041	0.850	0.092	0.149	0.004	0.084
TRM	Sudoku	0.117	0.213	0.198	0.294	0.091	0.104
TRM	ARC-AGI	0.586	0.693	0.401	0.016	0.011	0.025

Output-level readouts in full. Table 10 gives point estimates for every channel and readout, together with the unconditional flip rates and the achieved self-drop per cell. The same locality quantities with percentile-bootstrap intervals are below; flip locality is a conditional proportion over examples with at least one flipped target and is reported without an interval.

Model / task	Channel	State locality	Output-effect locality	Flip locality
HRM / Maze	within-L	0.265 [0.264, 0.266]	0.132 [0.130, 0.134]	0.278
	within-H	0.343 [0.340, 0.346]	0.372 [0.367, 0.376]	0.570
	cross-HH	0.220 [0.217, 0.222]	0.218 [0.215, 0.221]	0.360
HRM / Sudoku	within-L	0.266 [0.266, 0.267]	0.259 [0.258, 0.260]	0.240
	within-H	0.384 [0.383, 0.385]	0.367 [0.366, 0.368]	0.276
	cross-HH	0.339 [0.338, 0.339]	0.325 [0.324, 0.326]	0.263
HRM / ARC-AGI	within-L	0.477 [0.465, 0.490]	0.395 [0.380, 0.410]	0.442
	within-H	0.578 [0.565, 0.592]	0.560 [0.547, 0.574]	0.674
	cross-HH	0.527 [0.513, 0.541]	0.508 [0.495, 0.522]	0.617
TRM / Maze	within-L	0.032 [0.032, 0.032]	0.028 [0.028, 0.029]	0.041
	within-H	0.263 [0.261, 0.265]	0.266 [0.264, 0.269]	0.850
	cross-HH	0.060 [0.059, 0.061]	0.057 [0.057, 0.058]	0.092
TRM / Sudoku	within-L	0.119 [0.119, 0.119]	0.115 [0.115, 0.115]	0.117
	within-H	0.194 [0.194, 0.195]	0.211 [0.210, 0.212]	0.213
	cross-HH	0.209 [0.207, 0.210]	0.205 [0.204, 0.207]	0.198
TRM / ARC-AGI	within-L	0.631 [0.620, 0.643]	0.524 [0.509, 0.538]	0.586
	within-H	0.687 [0.674, 0.700]	0.655 [0.642, 0.670]	0.693
	cross-HH	0.455 [0.441, 0.470]	0.441 [0.426, 0.457]	0.401

D.5 Additional finite-noise activation patching results

Figures 11 and 12 show why reliability diagnostics are needed. In HRM/Maze, direct within-H perturbations are concentrated and reliable: H-level writes local summaries at each token. Perturbations carried by z_H across cycles decay more slowly, showing that the H state can propagate those

summaries more broadly. TRM/Maze contrasts with an almost baseline within-L channel under the shared recursive operator. Sudoku has higher locality at both levels because a 3×3 house already forms a broad constraint unit, but it preserves the same within-H direction.

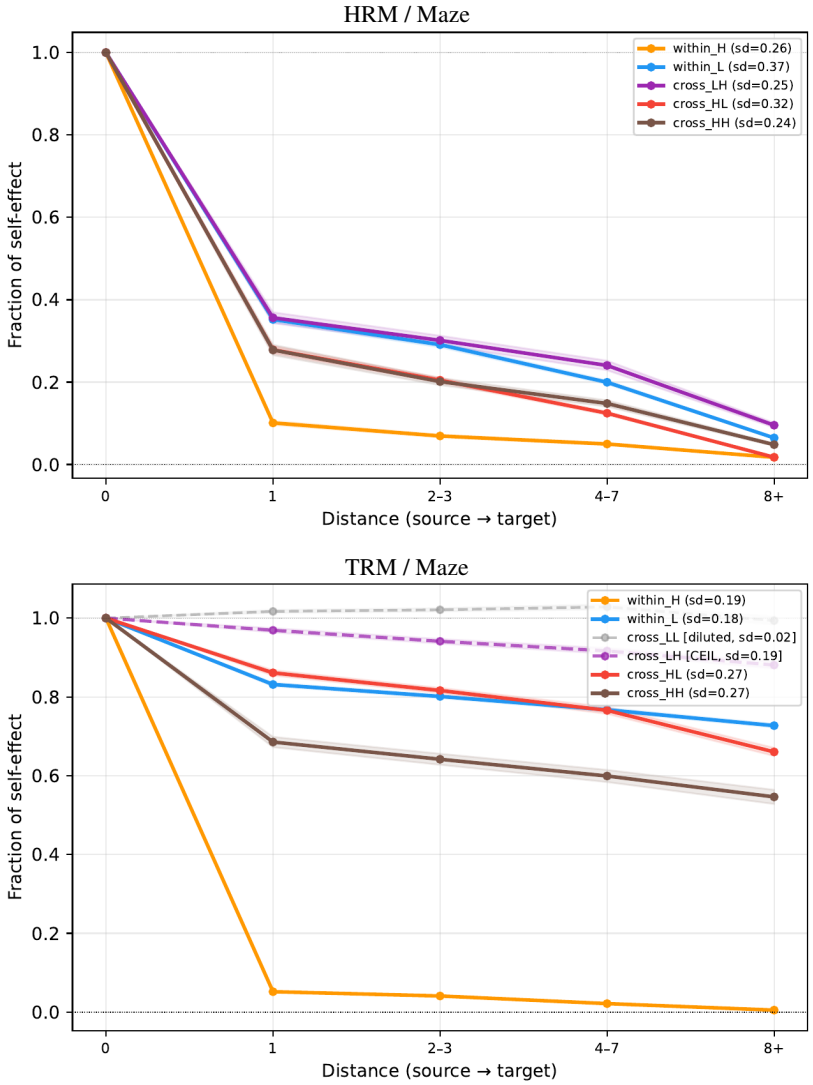


Figure 11: Normalized spatial decay of the self-effect in finite-noise activation patching on Maze, as the patch source moves from the target cell to farther cells. Colors denote within-H, within-L, and cross-level/cross-cycle source–target channels, and each legend entry gives the channel's self-drop (sd). HRM separates local H writes from broader cross-cycle H-state propagation, while TRM has a flatter shared-operator channel. Self-drop reliability for the same channels is the right column of Figure 13; a channel below 0.1 there has little effect on its own source prediction.

D.6 Constraint-type breakdown: comparing Jacobian, patching, and SAE ablation

The three probe methods in our framework—structural Jacobian, finite-noise activation patching, and SAE feature ablation—each make a different commitment to causal strength, granularity, and scope. This section places them side by side on the same question for Sudoku: *does information flow respect Sudoku constraint boundaries (box, row, column, or other)?*

For each cell pair (i, j) in the 9×9 grid we assign a constraint label based on which Sudoku rule links them: *box* if they share the same 3×3 subgrid (priority), *row*/*column* if they share

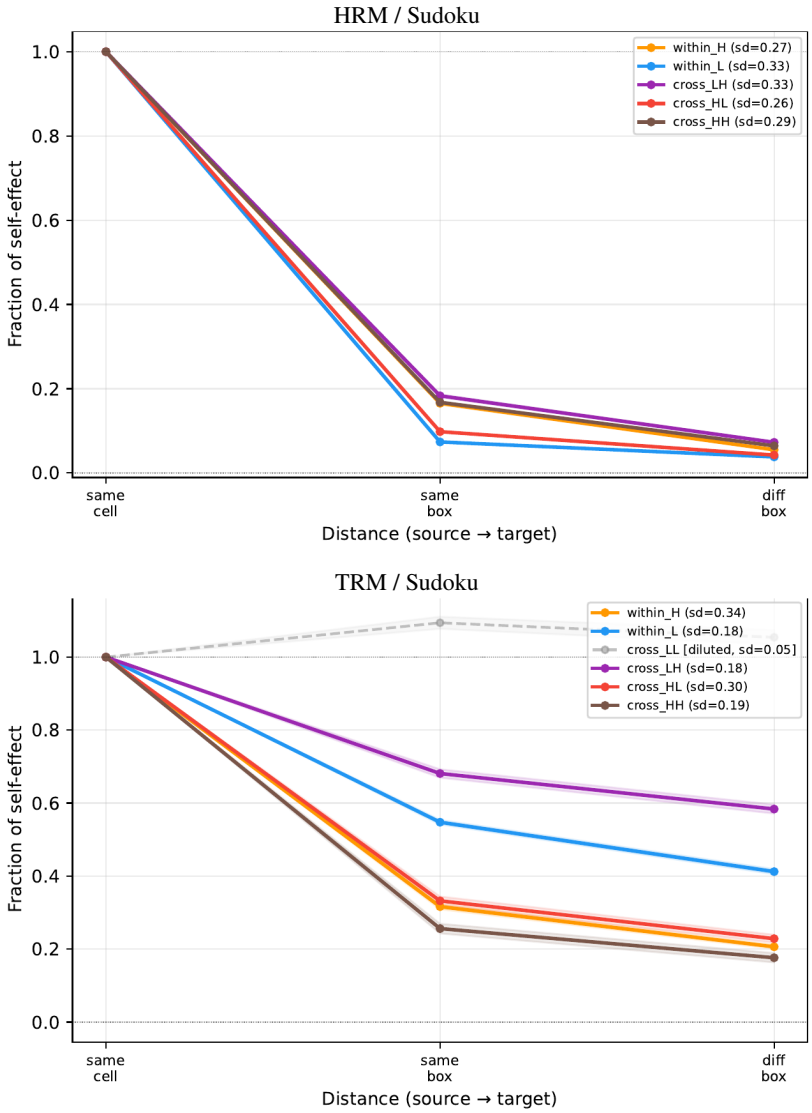


Figure 12: The spatial decay of Figure 11 on Sudoku, with distance measured as same cell, same box and different box. Colors and legend entries as there.

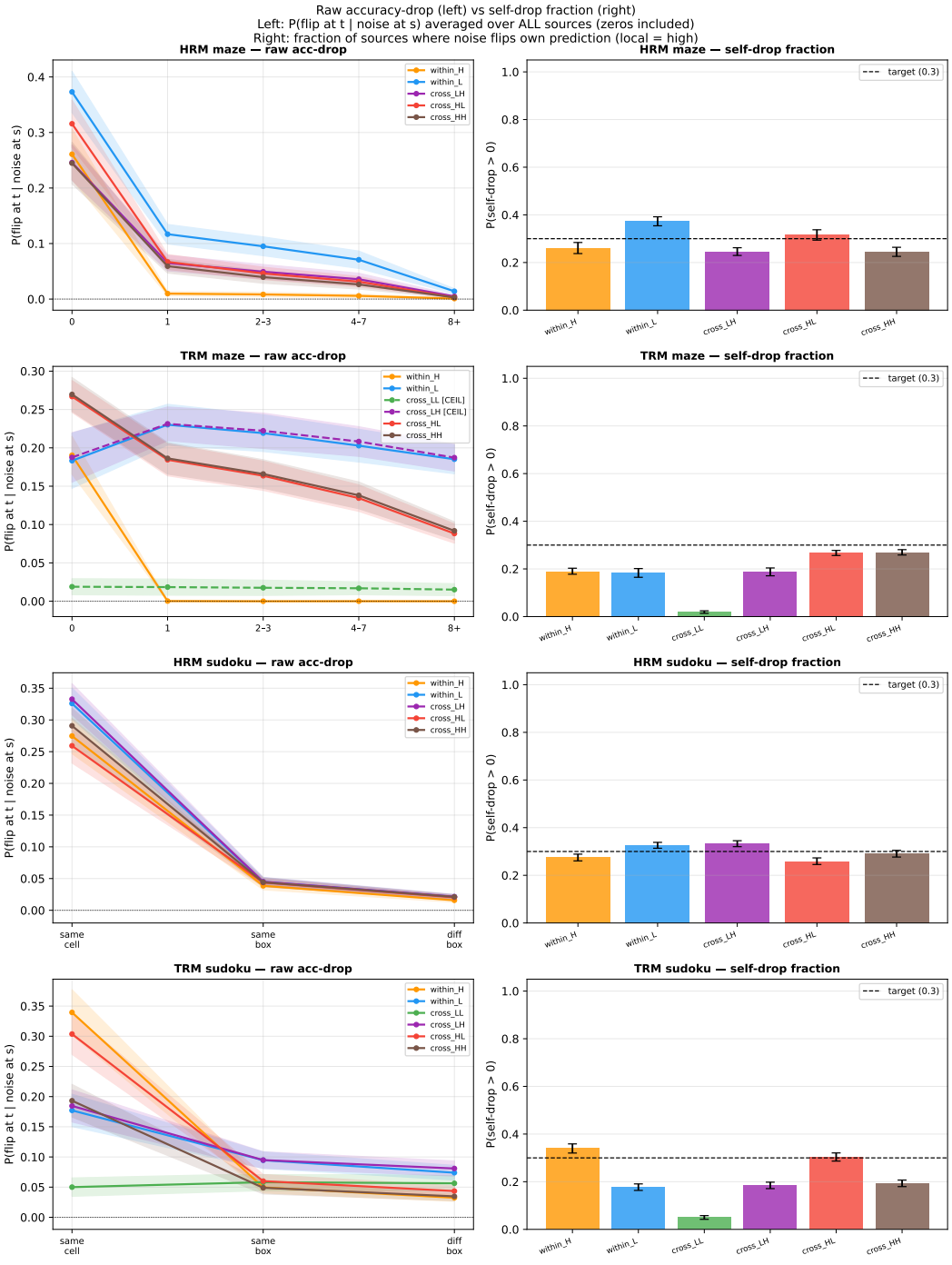


Figure 13: Raw accuracy-drop curves and self-drop fractions for the finite-noise activation patching. Raw drops distinguish genuine spatial reach from globally diluted channels that can look misleading after diagonal normalization.

a row/column outside the same box, and *other* otherwise. The random-assignment baseline is box = 0.100, row = 0.075, col = 0.075, other = 0.750 (exact counts from the 81×80 off-diagonal pair set).

Jacobian (structural): Per-cell pair, we compute the Frobenius-norm Jacobian $K[i, j] = \|\partial z[i]/\partial z[j]\|_F$ at the critical cycle, giving an (81,81) matrix per sample; fractions are sample-means weighted by total mass. The Jacobian figure covering HRM and TRM Sudoku is shown in Figure 18 (a).

Activation patching (causal): We use the averaged (81×81) heatmaps from the Jacobian-analog patching experiment; each entry is the mean impact on cell j when cell i is source-patched. Fractions are computed off-diagonal; 95% CIs use pair-resampling bootstrap over all 6480 off-diagonal pairs. Results are shown for the K_H- and K_L-analog channels (within-H and within-L, respectively).

SAE feature ablation (causal, feature-resolved): For the top-30 H-level and L-level SAE features (by total ablation impact), we zero each feature and record the change in output logits across all non-clue target cells. Impact is summed by constraint type; fractions average over features and samples.

Figure 14 presents all three analyses side by side for HRM (top row) and TRM (bottom row). The structural Jacobian K_H shows strong constraint-type preferences—in particular, K_L concentrates substantially on *column* pairs (≈ 0.25 vs. baseline 0.075) in HRM—while both activation patching and SAE feature ablation remain near the random baseline across all constraint types. This dissociation is informative: it shows that the cell-to-cell differential topology aligns with Sudoku structure, but individual sparse features and finite perturbations do not selectively activate along row/column lines. The implication is that constraint-type selectivity in Sudoku is a collective property of the full representational geometry rather than the province of individually identifiable sparse features or activation-patching channels.

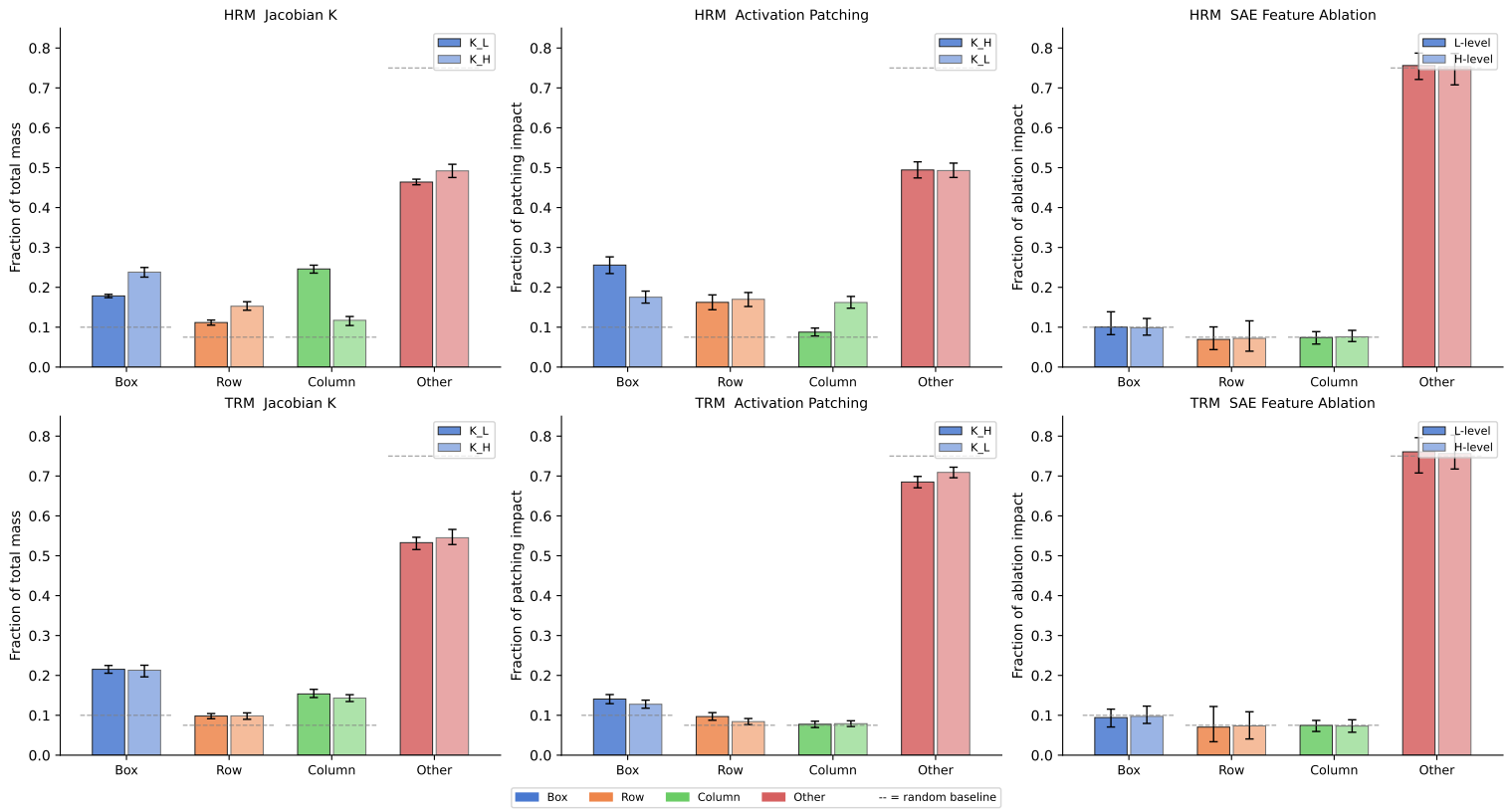


Figure 14: Sudoku constraint-type breakdown for all three probe methods, HRM (top) and TRM (bottom). **Col. 1 (Jacobian K):** structural Jacobian mass fraction by constraint type at the critical cycle; bars show K_L (darker) and K_H (lighter), error bars are 95% bootstrap CI over samples. **Col. 2 (Activation patching):** off-diagonal fraction of the averaged impact heatmap by constraint type; K_H- and K_L-analog channels; error bars from pair-resampling bootstrap. **Col. 3 (SAE feature ablation):** mean constraint-type fraction across the top-30 H-level and L-level features; error bars from sample-resampling bootstrap. Dashed lines mark the random pair-type baseline (box = 0.10, row = col = 0.075, other = 0.75). The Jacobian shows clear constraint-type structure (especially column concentration in K_L), while patching and SAE ablation are near-baseline—indicating that Sudoku constraint-type selectivity is a collective geometric property rather than a signature of individual sparse features or finite perturbation channels.

Table 11: Cycle-boundary locality and the boundary/within-cycle ratio, matched at six cycles per model, $n = 100$ mazes. All three models converge, so convergence does not require weight-tied recursion — a re-enterable carry suffices, and the non-recursive baseline has one. What separates them is the ratio: TRM's crosses 1.0 and locks near 1.15, while HRM's and the baseline's stay below 1 and barely move. Percentile bootstrap over examples, 2000 draws, seed 42. The baseline's ratio is an analogue, since it has no high/low pair.

Model	Quantity	Cycle					
		1	2	3	4	5	6
TRM	boundary	.560	.251	.251	.257	.259	.264
	ratio	.862	.634	1.162	1.186	1.157	1.143
HRM	boundary	.233	.128	.067	.064	.072	.080
	ratio	.447	.383	.424	.478	.493	.494
Non-recursive baseline	boundary	.164	.151	.154	.162	.164	.165
	ratio	.338	.520	.581	.612	.607	.608

E Recursion and depth

E.1 Recursion control

Table 11 gives the six-cycle convergence probe, Table 12 the endpoint comparison across the baseline's training trajectory, and Table 13 its sensitivity to the calibration target.

E.2 Protocol

The endpoint comparison across baseline training. The endpoint comparison reverses as the baseline trains, and the reversal is the opposite of what the undertrained comparison suggested. At the earliest checkpoint, and at the small sample the early checkpoints support, the recursive model appears to spread a source cell's effect more broadly. That ordering does not survive training: from roughly 5,500 cumulative epochs onward the non-recursive baseline spreads its effect *more* widely than TRM, and at its accuracy plateau the gap is $-.064$ with non-overlapping intervals. Read in the direction of the metric, the mature comparison says that the recursive model keeps a source cell's influence more tightly confined than a stack of distinct layers does. The sign is negative in all six calibrations of Table 13, with a monotone dose-response: the separation is largest under the gentlest perturbation ($-.076$ at self-drop .10) and shrinks as σ grows ($-.034$ at .50), consistent with large perturbations driving both architectures toward global mixing. The two matched-SNR rows at 0.5 and 1 sit near the geometry-matched null and we treat them as uninformative rather than as weak support.

Locality moves independently of competence. Two readings fit the endpoint ordering: recursion confines causal reach, or a less converged model simply has broader, sloppier reach. TRM's own training trajectory separates them. Across the four checkpoints preceding its late transition, cell accuracy improves from .966 to .983 while within-H locality *falls* from .069 to .040, the opposite of the competence account's prediction; across the transition, accuracy is flat (.983 to .983) while locality rises roughly sevenfold. Locality therefore varies independently of accuracy in both segments, and the architectural reading is the one the trajectory supports. We rest it on the trajectory and the depth profiles below rather than on the endpoint magnitudes.

What separates the architectures is how reach develops with depth. Figure 2 compares how locality evolves with computational depth rather than comparing endpoint magnitudes across models, which makes it robust to the accuracy gap. Both architectures start local and globalize. They end differently: the non-recursive baseline decays monotonically and is still decaying when it runs out of layers, with a terminal slope of $-.0389$ $[-.0406, -.0372]$, whereas TRM has stopped: fitting the low-level applications of its final cycle gives $-.0011$ $[-.0024, +.0002]$ per step. The two intervals are disjoint. TRM has stopped spreading: within its final cycle the fit is flat and the paired adjacent steps are mildly positive ($+.0020$ and $+.0038$), an order of magnitude below the baseline's decay either way. Measured over a wider window that spans TRM's last two low-level states and its high-

Table 12: Recursion versus depth on the shared input→output edge (Maze), as a function of how long the non-recursive baseline has trained. Locality is the fraction of change mass inside the path neighborhood, so a higher value means a *narrower* causal reach; the gap is the baseline minus TRM, so a negative gap means the non-recursive model spreads its effect more widely. $n = 300$ common-correct mazes per row, σ calibrated per model to a self-drop of .30. Epoch counts are cumulative, since the baseline was warm-started from an earlier run. Accuracy columns are over $n = 500$ mazes. TRM is constant across rows because only the baseline checkpoint varies.

Non-recursive baseline, by cumulative training epoch					
Cum. ep.	Path	EM	Locality [95% CI]	TRM	Gap
5,000	95.8%	0.366	0.209 [0.206, 0.211]	0.189	+0.020
5,500	95.8%	0.376	0.168 [0.166, 0.170]	0.189	−0.021
6,000	95.9%	0.378	0.169 [0.167, 0.171]	0.189	−0.020
7,000	96.2%	0.446	0.174 [0.171, 0.176]	0.189	−0.015
8,500	96.3%	0.466	0.140 [0.138, 0.142]	0.189	−0.049
11,000	96.4%	0.506	0.126 [0.125, 0.128]	0.189	−0.063
15,499	96.3%	0.508	0.118 [0.116, 0.120]	0.189	−0.071
24,499	96.2%	0.492	0.125 [0.122, 0.127]	0.189	−0.064
TRM `step_195310`: path acc. 98.0%, exact match 0.730, locality 0.189 [0.185, 0.194]. Geometry-matched baseline: 0.026.					

Table 13: Calibration sensitivity of the recursion comparison at the baseline's plateau checkpoint, $n = 300$. The gap is the non-recursive baseline minus TRM at each calibration; it is negative at every level, so the baseline spreads its effect more widely throughout, with the separation largest under the gentlest perturbation. The two matched-SNR rows at 0.5 and 1 fall close to the geometry-matched null of .026 and are treated as uninformative rather than as supporting evidence.

Calibration	Scale	σ (TRM / base)	TRM	Non-recurs.	Gap	
Self-drop	.10	0.108 / 0.059	0.154	0.078	−0.076	
Self-drop	.30	0.146 / 0.119	0.189	0.125	−0.064	
Self-drop	.50	0.181 / 0.168	0.200	0.166	−0.034	
Matched SNR	0.5	0.013 / 0.007	0.043	0.030	−0.012	near null
Matched SNR	1	0.027 / 0.014	0.059	0.037	−0.022	near null
Matched SNR	2	0.053 / 0.028	0.081	0.057	−0.025	

level write the slope reads $+.0199$, but that window crosses a role change and the level step between roles contributes most of it, so we report the single-role window (Table 14). The cycle-boundary series says the same thing: z_H locality drops from .567 to .251 and then does not move, the paired difference between the last two cycles being $+.0005$ $[-.0096, +.0112]$ (Table 16).

Settling is a property of the converged phase, not of recursion as such. HRM, the paper's other recursive model, is still spreading at the end of this probe: its final-cycle low-level slope is $-.1194$ $[-.1217, -.1171]$ per step, or $-.0299$ per layer against the baseline's $-.0389$. Read against cycle index rather than against the model label, this is what the TRM curve also does early on. TRM's plateau is a *third*-cycle phenomenon — its first transition falls steeply and only its second is flat — and HRM runs two cycles per adaptive-computation step, so the single transition observable here is the analogue of TRM's first, which likewise falls (Table 15). Settling therefore tracks how far the recursion has run rather than whether the architecture is recursive, and testing it requires matching cycle counts, which the next probe does. One further reason to compare cycle indices rather than magnitudes: a depth step is one transformer layer for the baseline, two for TRM and four for HRM, so signs and scale-invariant quantities carry across models while per-step magnitudes do not.

Convergence comes from state re-entry; reorganization comes from weight tying. Re-entering the carry removes the cycle-count asymmetry and lets all three models be matched at six cycles (Table 11). Under that probe *all three* converge, including the non-recursive baseline, which settles by its second cycle and then holds. The baseline is non-recursive only within an adaptive-computation step; it still carries state across steps, and that re-enterable carry is sufficient for convergence. Con-

Table 14: Terminal slope of locality with computational depth, on the shared input→output edge (Maze). Negative means the model is still spreading when it runs out of depth; non-negative means it has settled. Because a depth step is one transformer layer for the baseline, two for TRM and four for HRM, per-step magnitudes are not commensurate across models and the per-layer column is given alongside; the *sign* is invariant to that rescaling and carries the claim. Rows marked role-confounded fit across an $L \to H$ role change, which contributes a level step rather than a trend. Percentile-bootstrap 95% CIs over examples, 2000 draws, seed 42.

Model	Depth	Lay.	Slope/step	95% CI	/layer	
Non-recursive	8 layers	1	$-.0389$	$[-.0406, -.0372]$	$-.0389$	
TRM (role-mixed window)	15 unroll.	2	$+.0199$	$[+.0161, +.0238]$	$+.0100$	role-confounded
TRM (final-cyc. L)	15 unroll.	2	$-.0011$	$[-.0024, +.0002]$	$-.0006$	
HRM (all roles)	6 steps	4	$-.1321$	$[-.1336, -.1306]$	$-.0330$	role-confounded
HRM (final-cyc. L)	6 steps	4	$-.1194$	$[-.1217, -.1171]$	$-.0299$	

Table 15: Cycle-boundary z_H locality, aligned by cycle index. Differences are paired per example and bootstrapped over examples ($n = 100$, 2000 draws, seed 42). TRM's plateau is a third-cycle phenomenon: its first transition falls sharply and only its second is flat. HRM runs two cycles per ACT step, so its single observable transition is the analogue of TRM's first, and it likewise falls – by less. The probe measures one ACT step, so this comparison bounds how much can be concluded about HRM's convergence from curve shape alone.

Model	Transition	From	To	Paired Δ	95% CI	
TRM	cycle $1 \to 2$	.567	.251	$-.3160$	$[-.3200, -.3120]$	falling
TRM	cycle $2 \to 3$	.251	.252	$+.0005$	$[-.0096, +.0112]$	flat
HRM	cycle $1 \to 2$	.233	.128	$-.1058$	$[-.1095, -.1022]$	falling

vergence is therefore a property of iterated state re-entry rather than of weight tying, which locates it more precisely than the single-step probe allows. A further detail emerges only at six cycles: HRM's boundary series bottoms at its fourth cycle ($.064$) and turns back up to $.080$, so it is not a monotone decay toward the null.

What separates them is where they converge to. The plateaus differ in magnitude — TRM settles near .26, HRM near .08 and the baseline near .17 — but magnitude is exactly the quantity the differing layers per step contaminate. Figure 15 therefore reports the ratio of boundary-state to within-cycle locality, a scale-invariant quantity immune to that difference. Only TRM *reorganizes* that balance: it moves $0.86 \to 0.63$, crosses 1.0, and locks near 1.15 from the third cycle onward, holding across the adaptive-computation boundary. HRM settles at 0.38–0.49 and the baseline at 0.34–0.61, and neither crosses 1; iteration rescales the magnitude of their effects and leaves the high-relative-to-low structure alone. The dissociation we can support is therefore *whether iteration reorganizes the balance or merely rescales it*, and it now rests on two non-reorganizing comparisons, one of them genuinely non-recursive. As a control, TRM's first adaptive-computation step reproduces the single-step measurement to three decimal places, so re-entry does not manufacture the effect. The baseline has no high/low pair, so its ratio is an analogue built the same way — the last layer of each step over the preceding ones — rather than the same object.

Where the difference plausibly comes from. Full weight tying is the leading candidate: TRM reuses one operator for both roles, while HRM has distinct high-level and low-level modules and is weight-tied only across cycles, so HRM tests recursion-across-cycles rather than a single shared operator. Three other differences travel with it — cycles per step, layers per module, and puzzle-embedding length — so we name weight tying as the hypothesis the measurement points to and leave its isolation to a model set that varies these factors independently.

Reading across models. Two conventions carry across the comparison and one does not. Signs and scale-invariant ratios are comparable, because they are unaffected by the differing layers per depth step. Absolute per-step magnitudes are not, and neither are magnitudes across the two edges

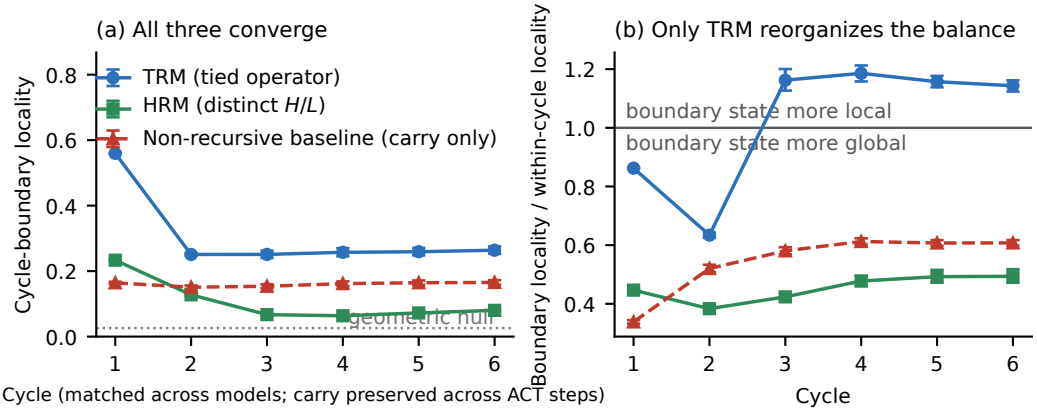


Figure 15: Carry re-entered so that all three models are matched at six cycles, $n = 100$, Maze. **(a)** All three converge, including the non-recursive baseline, whose carry is re-enterable across adaptive-computation steps even though it has no inner cycle loop. Convergence therefore does not require weight-tied recursion. **(b)** What each converges to, as the ratio of boundary-state to within-cycle locality — scale-invariant, so unaffected by the differing layers per step. Only TRM crosses 1.0, locking near 1.15; HRM (0.38–0.49) and the baseline (0.34–0.61) settle below it. The baseline's ratio is an analogue, since it has no high/low pair. Error bars are percentile-bootstrap 95% CIs over examples. The probe is fixed-depth and ignores adaptive-computation halting by design.

used here, since the trajectory evidence is read on the internal within-H edge and the endpoint comparison on the input→output edge. Section 7 states the design's remaining limits.

Protocol. Protocol for the comparison of Section 4.3. Source: `additional_experiments/recursion_control/`.

- **Baseline architecture and training.** `transformers_baseline@Model_ACTV2` with `H_cycles` $= 1$, eight layers, hidden width 512, configuration recovered field-for-field from the original baseline run and retaining the task, tokenizer, and adaptive-computation-time wrapper. Warm-started from `step_35154` and trained for a further 10h21m, yielding 40 checkpoints from `step_3906` to `step_156240`; eight geometrically spaced checkpoints were evaluated.
- **Cumulative epoch labels.** The warm start reset the step counter, so epochs are reported as $4500 + \text{step}/7.8125$. Raw step numbers would read as training from scratch.
- **Competence.** Path-cell accuracy and exact match at $n = 500$ for TRM and every baseline checkpoint appear in Table 12. The baseline plateaus at 96.2–96.5% and stops improving after roughly 11,000 cumulative epochs, never reaching TRM's 98.0%, so the intended 0.5 pt matching criterion is not attainable and we report both accuracies instead.
- **Common-correct set.** Each row uses mazes solved by both models, capped at $n = 300$.
- **Per-model calibration.** σ is fitted separately per model per calibration target; values appear in Table 13.
- **Buildup.** Per-layer locality for the baseline, per-application locality for TRM's fifteen unrolled module applications (four low-level cycles and the write per phase), and per-cycle z_H locality, with last-three-step slopes bootstrapped over examples (2000 draws, $n = 100$).

Artifact provenance. Every row of Table 12 is produced by one evaluation artifact per baseline checkpoint, named by checkpoint step rather than by a shared canonical path, so that the mature-checkpoint result cannot overwrite the intermediate ones. The released code records the checkpoint step, the fitted σ for each model, and the achieved common-correct sample size alongside each locality value.

Table 16: TRM's high-level state z_H at each cycle boundary on the input→output edge (Maze). Locality is the fraction of change mass inside the path neighborhood, so a higher value means a *narrower* causal reach. The series drops sharply after the first cycle and is then flat: the paired difference between the last two cycles straddles zero, which is what "settles" requires. Differences are computed *per example* and then bootstrapped, not as a difference of marginal means. $n = 100$ mazes, 2000 percentile-bootstrap draws, seed 42, resampling unit = example. This is the cycle-boundary series; the within-cycle trajectory is Figure 2.

Cycle	z_H locality	95% CI
1 of 3	.567	[.565, .569]
2 of 3	.251	[.247, .255]
3 of 3	.252	[.241, .263]
Paired differences (per example)		
Cycle 1 → 2	−.316	[−.320, −.312]
Cycle 2 → 3	.001	[−.010, .011] flat

Table 17: The two internal axes behave differently. $\Delta = L_{\text{within-H}} - L_{\text{within-L}}$, paired per example at $n = 500$. Across ACT reasoning steps Δ rises and saturates by step 8 in every cell – a convergence signature. Across internal h-cycles there is no such trend: the first cycle is the weakest in only three of six cells. The single negative Δ in the suite, HRM/Maze at C_0, is the earliest computational moment in that model (step 0, cycle 0) and is already positive by its next cycle.

		h-cycle axis		ACT step axis		
Model	Task	C_0	final C	S_1	S_{15}	$S_1 \to S_{15}$
HRM	Maze	−.0548	+.0777	+.0718	+.0899	+.0181
TRM	Maze	+.2851	+.2306	+.2633	+.2643	+.0010
HRM	Sudoku	+.1430	+.1177	+.1620	+.1817	+.0197
TRM	Sudoku	+.0307	+.0753	+.0809	+.0914	+.0105
HRM	Arc	+.0806	+.1007	+.1109	+.1137	+.0027
TRM	Arc	+.1937	+.0554	+.0855	+.0869	+.0014

Why MTU3D is not a third row. MTU3D's primary measurement is a deterministic zero ablation of an object's multi-view features read at the unified encoder's output, so it does not share the calibrated-noise input→output edge used here, and we report it as a geometry contrast in Section 4.4 rather than placing it on the recursion axis. What separates the two settings is the intervention rather than the readout: Section 4.4 reports grounding-logit and predicted-class readouts from the same forward passes, but a deterministic ablation of input features is not a behaviorally calibrated perturbation of a sited activation, and the neighborhood, units and null all differ.

E.3 Cycle-boundary series and the two internal axes

Table 16 gives TRM's cycle-boundary series on the input→output edge, and Table 17 reports the two internal axes, reasoning steps and h-cycles, separately because they behave differently.

E.4 Per-layer reading granularity

Each reasoning module contains several transformer layers, four in each of HRM's modules, two in TRM's, and one per layer in the baseline. Hooking the inner layers reads the same trajectory at finer granularity, giving 8 points for the baseline, 24 for HRM and 30 for TRM within one adaptive-computation step.

E.5 Additional cross-cycle Jacobian results

For cross-cycle kernels, diagonal concentration means the same row-normalized same-position mass $\text{mean}_u\, K[u,u] / \sum_v K[u,v]$ computed between aligned positions in adjacent recursive states. The

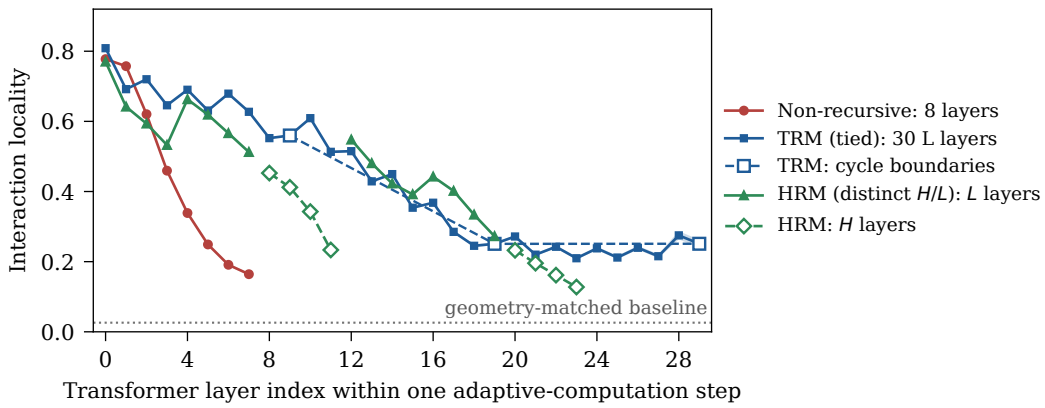


Figure 16: Locality after each transformer layer within one adaptive-computation step, Maze, $n = 100$. Color denotes model and line style denotes role: solid for low-level layers, dashed with open markers for HRM's two high-level blocks and for TRM's cycle boundaries. HRM's low-level segments are drawn separately either side of each high-level block, since no computation connects them directly. 95% bootstrap CIs over examples are mostly narrower than the lines (mean width .006, at most .024). The three models run 8, 24 and 30 layers per step, so a fixed-layer window is a different fraction of each model's depth; signs and shapes are read across models and magnitudes within one.

tables report cross-cycle locality summaries and the same-position concentration where H-to-H terms are available.

Table 18: Cross-cycle Jacobian locality by cycle edge (mean [95% CI], $n = 100$ per cell). For HRM, $H0L0 \to H0L1$ is the within-first-H-step update, $H0L1 \to H1L0$ is the cross-H-step boundary, and $H1L0 \to H1L1$ is the within-second-H-step update. TRM rows use analogous late-within-H and boundary anchors, with exact edge labels shown because L-cycle counts differ across tasks.

Architecture	Task	Cycle edge	Cross-cycle locality [95% CI]
HRM	Maze	$H0L0 \to H0L1$	.263 [.257, .270]
HRM	Maze	$H0L1 \to H1L0$	.261 [.256, .268]
HRM	Maze	$H1L0 \to H1L1$	.260 [.254, .266]
HRM	Sudoku	$H0L0 \to H0L1$	.264 [.263, .265]
HRM	Sudoku	$H0L1 \to H1L0$	.268 [.267, .268]
HRM	Sudoku	$H1L0 \to H1L1$	.243 [.242, .243]
HRM	ARC-AGI	$H0L0 \to H0L1$	.380 [.349, .414]
HRM	ARC-AGI	$H0L1 \to H1L0$	.380 [.350, .413]
HRM	ARC-AGI	$H1L0 \to H1L1$	.378 [.347, .411]
TRM	Maze	$H0L2 \to H0L3$	.501 [.499, .504]
TRM	Maze	$H0L3 \to H1L0$	.501 [.499, .503]
TRM	Maze	$H1L2 \to H1L3$	.499 [.497, .502]
TRM	Sudoku	$H0L4 \to H0L5$	.326 [.325, .328]
TRM	Sudoku	$H0_{z_H} \to H1L0$	.326 [.325, .328]
TRM	Sudoku	$H1L4 \to H1L5$	.326 [.324, .328]
TRM	ARC-AGI	$H0L2 \to H0L3$	.513 [.485, .542]
TRM	ARC-AGI	$H0L3 \to H1L0$	.514 [.486, .543]
TRM	ARC-AGI	$H1L2 \to H1L3$	.513 [.485, .541]

Table 19 gives additional quantitative detail for the appendix structural cross-cycle analysis.

F Structural and feature-level probes

F.1 Additional SAE locality results

Figure 17 shows two representative SAE features, and Table 20 compares SAE feature locality with Jacobian locality per position.

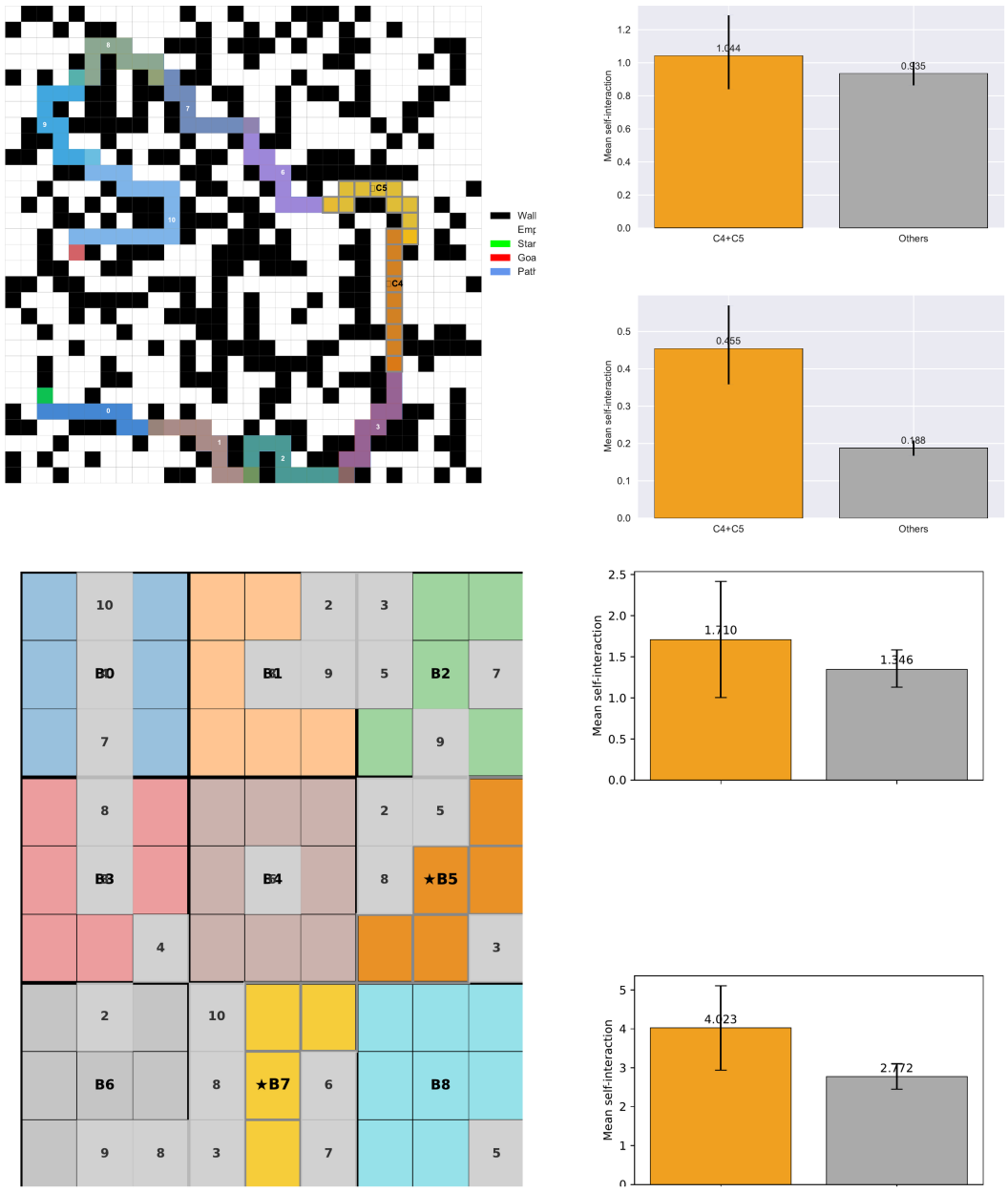


Figure 17: Representative SAE semantic locality examples. Top: a Maze HRM feature whose ablation concentrates on a corridor pair; the colored corridor overlay marks the selected segment and the bar plots compare that segment with all other path cells. Bottom: a Sudoku HRM feature whose ablation concentrates on a pair of 3×3 boxes; orange bars denote the highlighted box pair and gray bars denote the comparison set, with error bars showing 95% CIs. These feature ablations give interpretable segment-level hypotheses, such as corridors and Sudoku boxes; causal reach is tested by the finite-noise patching results in Table 1 and Figure 11.

Table 19: Available $H \to H$ cross-cycle diagonal concentrations (mean [95% CI], $n = 100$ per cell). Values are listed as sequences over H-update pairs when present in the updated file. Maze and Sudoku TRM have no comparable $H \to H$ term under the analyzed L-cycle indexing, while ARC-AGI TRM has z_H boundary entries.

Architecture	Task	$H \to H$ diagonal concentration sequence
HRM	Maze	.900 [.899, .902] → .462 [.452, .474]
HRM	Sudoku	.480 [.480, .480] → .218 [.216, .220]
HRM	ARC-AGI	.736 [.698, .775] → .650 [.604, .696] → .517 [.461, .573] → .507 [.450, .563]
TRM	Maze	–
TRM	Sudoku	–
TRM	ARC-AGI	.754 [.724, .781] → .659 [.623, .693] → .602 [.560, .642]

Table 20: Per-position correspondence between SAE feature locality and Jacobian locality. Correlations are computed from the cross-dataset summary files. Correlation is task- and architecture-dependent, reinforcing that SAE segment locality and Jacobian cell-to-cell locality are complementary rather than interchangeable measurements.

Configuration	n	Pearson r	Spearman ρ
Maze HRM	10966	.008 [n/a]	.009 [n/a]
Maze TRM	114	.016 [-.126, .207]	.118 [-.073, .310]
Sudoku HRM	8100	.478 [.459, .496]	.393 [.373, .412]
Sudoku TRM	8100	.026 [.005, .047]	.021 [-.000, .043]
ARC-AGI HRM	2652	.152 [.111, .193]	.140 [.102, .175]
ARC-AGI TRM	1768	.510 [.471, .546]	.501 [.461, .538]

F.2 Additional within-cycle Jacobian and intervention results

The finite-noise patching summary in Table 1 is the primary within-cycle causal evidence in the main text. The Jacobian results below are structural diagnostics: they summarize linearized cell-to-cell and segment-level topology, and help explain why the finite-intervention patterns arise, but are not treated as large-intervention causal effects.

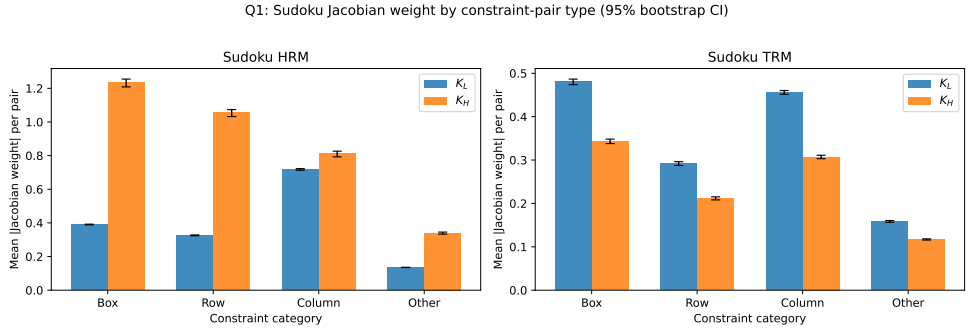
For a Jacobian diagnostic, we write $K^{b \leftarrow a}[u, v] = \|\partial z_b[u] / \partial z_a[v]\|_F$ for the local linear influence from source site v at level a to target site u at level b. The shorthand K_L and K_H in the appendix figures denotes the L-output and H-output kernels, respectively; it does not assume that K_L is local or that K_H is global. Cell locality is measured after the fact by the row-normalized diagonal concentration $\bar{\ell}(K) = \text{mean}_u K[u, u] / \sum_v K[u, v]$. For segment diagnostics, a partition $\mathcal{S} = \{S_m\}$ induces $K^{\text{seg}}[m, n] = \text{mean}_{u \in S_m, v \in S_n} K[u, v]$. When a segment-granularity score is reported in supplementary outputs, we use

$$g(K) = \frac{r_{\text{seg}}}{1 + \bar{\ell}(K)}, \tag{4}$$

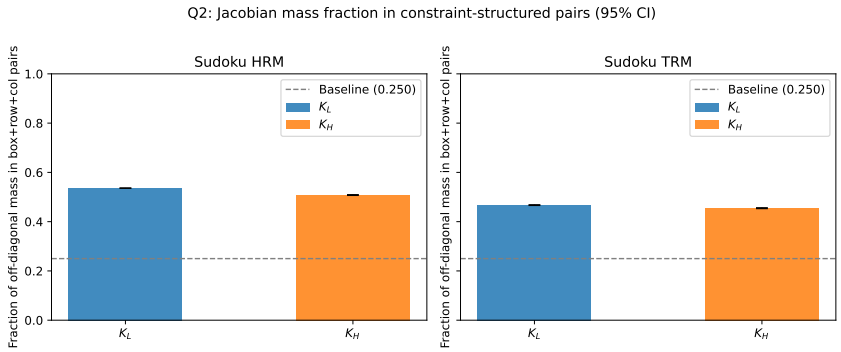
where r_{seg} is the mean same-segment entry divided by the mean cross-segment entry. The normalization keeps this descriptive segment score from simply duplicating cell-diagonal concentration.

Within-cycle structural locality. Table 21 restores the within-cycle Jacobian summary used as a structural check. In HRM, the L-output kernel is more same-position local than the H-output kernel in Maze and Sudoku, while ARC-AGI collapses the H/L gap under object-local geometry. TRM changes this ordering: Maze and ARC-AGI have larger H-output locality, while Sudoku is nearly tied. These linearized kernels are not the main finite-causal evidence, but they show that the task geometry is also visible in local differential topology.

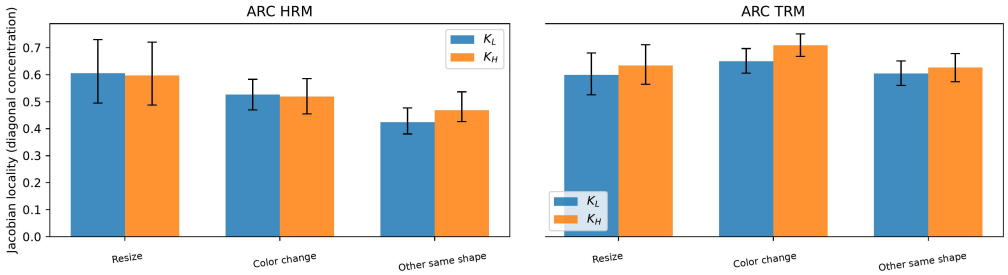
Segment granularity. Table 22 aggregates the same kernels into task segments. Maze HRM remains L-over-H at both cell and corridor resolutions, while Maze TRM is H-over-L; Sudoku is weaker but consistent with the nearly tied TRM pattern; ARC-AGI shows that object-level segments can reverse the HRM cell-level near tie. These results contextualize the main finite-noise patching results by showing where the structural kernels place their segment mass.



(a) Sudoku constraint-pair breakdown.



(b) Sudoku structured-pair mass fraction.



(c) ARC-AGI locality by task type.

Figure 18: Additional quantitative Jacobian segment analyses. Panel (a) breaks Sudoku Jacobian weight into box, row, column, and other cell-pair categories; panel (b) summarizes the off-diagonal mass in constraint-structured pairs; panel (c) stratifies ARC-AGI locality by task type. Segment-conditioned statistics expose structure that is not visible from the scalar cell-locality summary alone: HRM and TRM differ in how K_L and K_H distribute mass across Sudoku constraint types, and ARC-AGI locality varies with task type.

Table 21: Within-cycle Jacobian same-position locality summary (95% bootstrap CI). K_L and K_H are L-output and H-output Jacobian kernels, respectively; they are level-indexed structural diagnostics, not assumed local or global a priori. Sample sizes are $n = 191$ for HRM Maze, $n = 530$ for TRM Maze, and $n = 100$ for the remaining model–task pairs.

Architecture	Task	K_L locality [95% CI]	K_H locality [95% CI]	Ratio
HRM	Maze	.331 [.328, .333]	.225 [.220, .230]	1.47×
HRM	Sudoku	.289 [.288, .289]	.205 [.204, .206]	1.41×
HRM	ARC-AGI	.541 [.492, .594]	.537 [.483, .597]	1.01×
TRM	Maze	.576 [.575, .578]	.701 [.699, .704]	.82×
TRM	Sudoku	.372 [.371, .374]	.368 [.365, .370]	1.01×
TRM	ARC-AGI	.632 [.597, .673]	.681 [.647, .720]	.93×

Table 22: Segment-granularity summary with 95% bootstrap CIs. Granularity is the segment self/cross ratio normalized by cell-level locality as in Equation (4); higher values indicate influence concentrated within task segments but spread across cells inside those segments.

Architecture	Task	$g(K_L)$ [95% CI]	$g(K_H)$ [95% CI]	Direction
HRM	Maze	7.35 [7.29, 7.41]	4.35 [4.27, 4.43]	L > H
HRM	Sudoku	4.66 [4.62, 4.69]	3.99 [3.96, 4.03]	L > H
HRM	ARC-AGI	2.57 [2.30, 2.87]	3.17 [2.82, 3.51]	H > L
TRM	Maze	12.08 [12.02, 12.14]	15.60 [15.45, 15.74]	H > L
TRM	Sudoku	8.05 [7.96, 8.14]	7.84 [7.74, 7.98]	near tie
TRM	ARC-AGI	4.53 [4.10, 4.99]	6.29 [5.59, 7.07]	H > L

G MTU3D

G.1 MTU3D DEPTH PROFILE

Table 23 gives the query-slot probe of Section 4.4 at each read depth, with its layer-zero control.

G.2 RECOVERY LOCALITY ON MTU3D UNDER THREE SCORE DEFINITIONS

Section 4.4 reports recovery locality with the recovered mass unnormalised (S1). Table 24 places that definition beside the per-target normalised one (S2) and a guarded variant that drops the lowest-denominator targets (S3), all on the same 200 scenes. The three agree in direction and differ in level by a stable offset, which is the signature of a distance-dependent denominator: the per-target normaliser averages 1.84 for near objects against 0.99 for far ones, so dividing by it upweights far objects and depresses the near share by about .11 at every depth. Injection layer 0 is a no-op, since clean and corrupted encoder inputs are identical there, and is shown as undefined.

G.3 ADDITIONAL MTU3D LOCALITY RESULTS

We apply the framework to MTU3D, a large-scale 3D vision-language grounding model (Zhu et al., 2025), on ScanNet indoor scenes (Dai et al., 2017). The question is whether ablating one object-level scene feature changes nearby objects more than distant ones, with neighborhoods defined by metric object distance rather than grid adjacency. This is a geometry contrast rather than a recursion contrast: MTU3D differs from HRM and TRM in architecture, task, representation, and intervention site simultaneously, so it bounds where the framework applies without isolating any single cause.

Causal spatial locality is present, and it is present throughout. At the visual-to-grounding handoff, ablating one object's features gives locality $.524$ $[.497, .552]$ against a geometry-matched null of $.420$, and the effect stays above its baseline in every object-density stratum and for all four released checkpoints ($.524$, $.502$, $.527$, $.507$ against $.420$; Appendix G.3). The margin is smallest in low-density scenes ($.809$ against $.767$), where few objects leave the baseline nearly saturated and the measurement least informative, and largest in high-density scenes ($.299$ against $.158$). Reading the

Table 23: MTU3D locality by read depth from a local start, 200 one-per-room ScanNet scenes. The probe perturbs one object's *populated* query slot at layer 1; the control perturbs layer 0, where every object's slot is the same vector so no object-specific representation exists yet. Adjacent differences are paired per scene. Magnitudes are not comparable to the 2D curves — different perturbation, units and null — so read signs and shape only.

Injection	Read depth	Locality	95% CI	Paired step
Probe (layer 1)	layer 2	.616	[.595, .638]	
	layer 3	.587	[.563, .611]	−.0292 [-.0322, -.0262] decay
	layer 4	.589	[.565, .613]	+.0019 [+.0009, +.0030] rise
Control (layer 0)	layer 1	.552	[.523, .581]	
	layer 2	.559	[.533, .586]	+.0072 [+.0034, +.0110] rise
	layer 3	.550	[.524, .578]	−.0089 [-.0099, -.0079] decay
	layer 4	.552	[.526, .579]	+.0020 [+.0012, +.0028] rise
Geometry-matched baseline	—	.434	[.400, .470]	

Table 24: MTU3D layer-recovery locality by **injection** depth under three score definitions, 200 one-per-room ScanNet scenes. A clean object's layer-k input is patched into the corrupted run and the effect read at the final query space. S1 scores the recovery mass $\|q_{\text{patched}} - q_{\text{corrupt}}\|$; S2 divides it per target by $\|q_{\text{clean}} - q_{\text{corrupt}}\|$, the per-target normalised definition, which converts mass into per-object ratios and lets far objects with near-zero denominators dominate; S3 is S2 with the lowest-denominator targets dropped. Injection layer 0 is a provable no-op (clean and corrupt encoder inputs are bit-identical), so its share is undefined and reported as '—', not 0. † exceeds the geometry-matched null, 0.434 $[0.400, 0.470]$. Percentile bootstrap over scenes, 2000 draws, seed 42.

Score	inj. 0	inj. 1	inj. 2	inj. 3	L3−L1 (paired)
S1 unnorm. *(primary)*	—	0.589† [0.566, 0.613]	0.609† [0.587, 0.633]	0.649† [0.630, 0.669]	+0.060 [+0.055, +0.065]
S2 per-target normalised	—	0.477† [0.446, 0.509]	0.493† [0.463, 0.524]	0.528† [0.501, 0.557]	+0.051 [+0.047, +0.055]
S3 norm., guarded	—	0.515† [0.486, 0.547]	0.530† [0.501, 0.561]	0.563† [0.536, 0.591]	+0.047 [+0.043, +0.052]

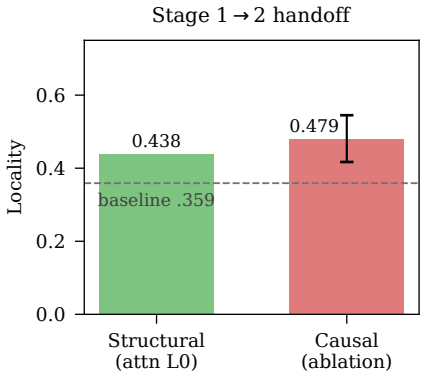


Figure 19: Interaction locality at the MTU3D visual-to-grounding handoff, 30-scene set. Structural layer-0 attention locality (.438) and causal input ablation (.479, 95% CI [.417,.545]) both exceed the geometry-matched near-object baseline of .359 (dashed). The 200-scene, density-stratified and checkpoint-stratified versions are in Appendix G.3.

same ablation at each unified-encoder layer rather than only at the handoff gives .501, .521, .520, .536 against a baseline of .434 — above the null at every depth (Table 26).

Two probes are used in this section, both in the ablation direction: an input-level zero-ablation of one object's multi-view features, read at the handoff and at each encoder layer, and a scaled perturbation of one object's populated query slot at layer 1, read at each subsequent layer. Neither restores a clean state into a corrupted run.

Table 25: MTU3D output-level ablation, 200 one-per-room scenes. The same intervention is read three ways on the same forward passes: the encoder's final state, the per-object grounding logit, and the per-object predicted class. State and output localities are paired per scene. This is the MTU3D counterpart of the HRM/TRM output-level checks, which show output and flip locality above the geometric null in all six cells. †Flip locality is conditioned on scenes with at least one off-source class flip: the locality of an all-zero flip vector is undefined and scores zero by convention, and most ablations produce no flip, so the unconditional mean measures how often flips occur rather than where they land.

Readout	Locality	Paired vs. baseline	
Encoder state	.536	$+.101$ $[+.094, +.108]$	above baseline
Grounding logit	.513	$+.078$ $[+.071, +.085]$	above baseline
Predicted-class flip†	.610	$+.222$ $[+.146, +.300]$	above baseline
Geometry-matched baseline	.434		
Output − state (paired)		$-.023$ $[-.028, -.018]$	output tracks state
Unconditional flip rate	.009		
Grounded-answer flip rate	.116		one global choice per scene

The effect reaches the model's output. The released forward pass returns per-object grounding and class logits, so the same output-level checks we apply to the grid tasks are available here. Read on the same forward passes as the state measurement, grounding-logit locality is .513 against the .434 baseline, a paired margin of $+.078$ $[+.071, +.085]$, and output tracks state to within .023 (Table 25). Ablating one object changes which object the model grounds in 11.6% of scenes. The MTU3D measurements are therefore behaviorally grounded in the same sense as the grid-task ones, not merely properties of a hidden state.

On the axis comparable to the 2D buildup, MTU3D spreads. Comparing depth profiles across the 2D and 3D settings requires perturbing the same kind of object. The 2D buildup perturbs one token's residual stream, whereas the ablation probe removes an object's input features before any object-specific representation exists: at layer 0 every object's query slot holds the same vector, with a mean pairwise distance of 0.01 across objects, and object identity first appears at layer 1, where that distance is 17.3. Perturbing one object's *populated* query slot at layer 1 is the matching intervention, and under it locality *decays* with read depth at $-.0136$ $[-.0152, -.0121]$, against a layer-0 control that is flat at $-.0035$ $[-.0041, -.0028]$; the paired probe-minus-control difference is $-.0102$ $[-.0115, -.0089]$ and holds across three perturbation scales (Table 23). On the comparable axis MTU3D therefore behaves like the 2D non-recursive baseline: it is still spreading at the end of the encoder, which places it with depth rather than with the settling recursive model.

This is one transition rather than a trend. The paired adjacent differences place essentially all the movement between the second and third read depths ($-.0292$), with the last step slightly positive ($+.0019$), and three read depths is the architectural ceiling given four encoder layers and injection at layer 1. We do not add a point at the injection site, where the off-source change is exactly zero and locality is undefined.

The defensible reading is therefore about where causal spatial structure sits and how it develops, not about its absence. MTU3D concentrates it most strongly at a module boundary, retains it at every encoder depth, carries it through to the grounded output, and spreads it with depth in the manner of a non-recursive 2D model. None of this identifies recursion as the cause of anything; the controlled comparison bearing on that question is the matched non-recursive baseline of Section 4.3.

Scene, density, and checkpoint conditions. Per-condition results for the visual-to-grounding handoff. Each condition carries its own geometry-matched baseline, because the expected local fraction depends on how many objects the scene contains and on the adaptive radius.

Table 26: MTU3D ablation locality by encoder read depth, 200 one-per-room ScanNet scenes. One object's multi-view features are zero-ablated at the input and the ℓ_2 change is read at the output of each unified-encoder layer under the per-scene metric radius (about seven neighbors on average). Locality exceeds the geometry-matched baseline at every depth. Percentile bootstrap over scenes, 2000 draws, seed 42.

	Baseline	Read at layer k			
		1	2	3	4
Ablation locality	.434	.501	.521	.520	.536

Condition	n	Causal locality [95% CI]	Baseline
Scene set (pretrain checkpoint)			
Original	30	0.479 [0.417, 0.545]	0.359
Expanded	200	0.524 [0.497, 0.552]	0.420
Object density (expanded set)			
Low ($<$ 15 objects)	58	0.809 [0.776, 0.839]	0.767
Medium (15–35)	96	0.459 [0.444, 0.476]	0.335
High ($\geq$ 35)	46	0.299 [0.287, 0.313]	0.158
Released checkpoint (expanded set)			
Pretrain	200	0.524 [0.497, 0.552]	0.420
OVON fine-tune	200	0.502 [0.474, 0.531]	0.420
GOAT fine-tune	200	0.527 [0.501, 0.555]	0.420
SG3D fine-tune	200	0.507 [0.479, 0.536]	0.420

Causal locality exceeds its geometry-matched baseline in every condition. The margin is smallest in low-density scenes, where few objects leave the baseline nearly saturated and the measurement is least informative, and largest in high-density scenes. Locality is stable across all four released checkpoints, so the handoff effect is not an artifact of a particular fine-tuning stage.

Table 26 reads the input ablation at each encoder layer.

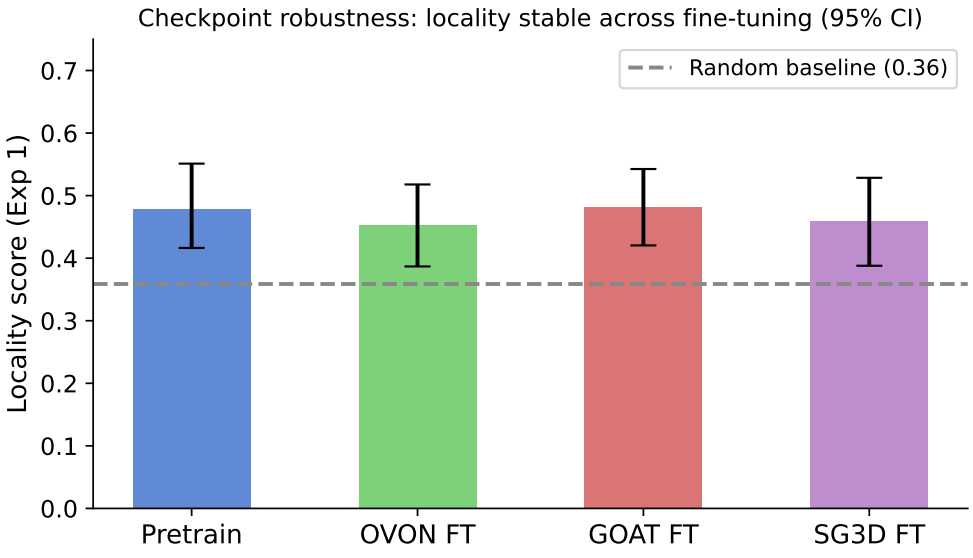


Figure 20: MTU3D locality across checkpoints. Stage 1–2 input-ablation locality is stable across checkpoints.

H CALIBRATION AND DISPLACEMENT

H.1 DISPLACEMENT OF THE PERTURBED STATE

Every analog has an inject site and a capture site, and the displacement at each is compared with the natural between-example spread of the clean state at that same site. At the inject site the displacement is $\|\sigma\epsilon\|$ exactly; at the capture site it is $\|z^{\text{pert}} - z^{\text{clean}}\|$ at the perturbed position, the diagonal of the effect matrix. The capture site is the one the paper reads. Cluster bootstrap over examples, 2000 draws, seed 42, each site compared with the spread at that same site; every one of the eighteen capture-site ratios resolves, eleven below one and seven above. Four notes travel with these numbers. Clean states are RMS-normalized to $\|z\| = \sqrt{512}$ at every site, so σ is a multiple of the state norm and ratios compare across cells directly. The cross-HH inject site on Maze and Sudoku is the high-level initialization, whose between-example spread is below 0.25 of the state norm, so its inject ratios are marked degenerate and are not comparable with the rest. For rows near one the choice of spread variant moves the ratio more than its interval does, so HRM/Maze within-L should be read as at the natural spread. TRM/Maze within-H at 5.3 is the largest displacement, driven by an unusually tight capture-site spread.

Table 27 gives every site and edge, and Table 28 separates the radial and angular components of the injected displacement. Because every block output is post-normed, the radial component is not a small correction: at $\sigma = 49$ the injected vector is 49 times the state it is added to, and the next block's attention and MLP consume that state before its own norm restores $\sqrt{d}$. In nine of the fourteen non-degenerate rows the perturbation also turns the state through a larger angle than two clean examples are apart, so it is not a local displacement of the model's own states even with the radial part set aside. Table 29 checks the achieved self-drop at $n = 200$: five of twelve analogs miss the 30% target with a 95% interval excluding it, four above and one below, and refitting the scale moves it by a median of 17%. The $\sigma \approx 49$ values on TRM's low-level states are not fit failures; an independent refit lands at 50.0 and 43.5, and those sites genuinely require a perturbation of that size to change 30% of predictions. Table 30 joins displacement to the six-setting scale sweep, which preserves the ordering in 26 of 36 cell–setting pairs at $n = 500$, 16 of 18 under self-drop with both exceptions in HRM/Sudoku and 10 of 18 under matched SNR, so the ordering is strongest where the intervention is furthest from the activation distribution.

Table 27: On-manifold displacement at both sites of each measured edge (§H.1). $\|\sigma\epsilon\|$ is the injected magnitude (median of actual draws); spread is the natural between-example distance at the *same* site. The **capture ratio** is the headline: displacement at the site the paper reads, over the spread there, with a cluster bootstrap CI over examples (2000 draws, seed 42). Inject-site intervals are marked † because $\|\sigma\epsilon\|$ has no example dimension, so only the denominator is resampled. Rows marked ‡ have an inject-site spread below $0.25\times$ the state norm: z_H at cycle $(0, 0, 0)$ barely varies across examples there, so there is little manifold to be off and the inject ratio must not be read alongside the others. The two ARC cross-HH rows are *not* marked — their inject spread is .41 and .71 of the state norm, compressed relative to the ≈ 1.3 at the other sites but a real distribution. The intervals hold the spread *variant* fixed, and for rows near 1 the choice of variant moves the ratio further than the interval width: HRM/Maze within-L (0.98 vs 1.05) crosses 1 under the other variant, so it should be read as *at* the natural spread rather than resolvably on one side of it.

cell	edge	σ	$\|\sigma\epsilon\|$	inj. spread	inject ratio	cap. spread	capture ratio
HRM/Maze	within-H	4.31	97.6	30.77	3.17 (3.16, 3.20)†	8.29	2.82 [2.69, 2.95]
HRM/Maze	within-L	6.22	140.3	15.00	9.36 (9.21, 9.51)†	30.77	0.98 [0.98, 0.99]
HRM/Maze	cross-HH	2.13	48.2	1.00	48.10 (47.00, 49.34)†‡	8.29	2.50 [2.39, 2.62]
TRM/Maze	within-H	15.28	346.4	23.33	14.85 (14.72, 15.04)†	5.28	5.33 [5.21, 5.46]
TRM/Maze	within-L	49.12	1108.0	23.95	46.27 (45.74, 46.93)†	23.33	1.39 [1.38, 1.41]
TRM/Maze	cross-HH	25.31	574.4	2.07	278.10 (269.77, 288.30)†‡	5.28	4.43 [4.33, 4.54]
HRM/Sudoku	within-H	3.13	71.1	26.13	2.72 (2.69, 2.75)†	31.08	0.79 [0.78, 0.79]
HRM/Sudoku	within-L	12.63	284.9	20.76	13.72 (13.30, 14.05)†	26.13	1.15 [1.14, 1.17]
HRM/Sudoku	cross-HH	1.87	42.5	1.94	21.97 (19.82, 25.61)†‡	31.08	0.86 [0.86, 0.87]
TRM/Sudoku	within-H	7.72	174.9	27.76	6.30 (6.26, 6.34)†	30.98	0.90 [0.89, 0.90]
TRM/Sudoku	within-L	49.19	1109.6	27.87	39.82 (39.53, 40.10)†	27.76	1.11 [1.11, 1.12]
TRM/Sudoku	cross-HH	4.27	96.8	4.69	20.65 (19.89, 21.44)†‡	30.98	0.80 [0.79, 0.81]
HRM/ARC	within-H	1.00	22.7	29.03	0.78 (0.78, 0.78)†	28.43	0.16 [0.16, 0.17]
HRM/ARC	within-L	1.00	22.6	28.91	0.78 (0.78, 0.78)†	29.03	0.14 [0.13, 0.14]
HRM/ARC	cross-HH	1.00	22.7	9.31	2.44 (2.36, 2.51)†	28.43	0.20 [0.19, 0.21]
TRM/ARC	within-H	1.00	22.7	29.56	0.77 (0.76, 0.77)†	17.55	0.28 [0.26, 0.29]
TRM/ARC	within-L	1.00	22.6	29.25	0.77 (0.77, 0.77)†	29.56	0.32 [0.31, 0.32]
TRM/ARC	cross-HH	1.00	22.7	16.11	1.41 (1.38, 1.44)†	17.55	0.18 [0.17, 0.19]

Table 28: Radial and angular decomposition of the injected displacement at the inject site. Every clean state is post-normed to $\|z\| = \sqrt{d} = 22.627$ (`rms_norm` on every block output), so the noise-to-signal ratio $\|\sigma\epsilon\|/\|z\|$ *equals* σ, the perturbed norm is $\sqrt{1+\sigma^2}$ times the clean norm, and chord distances convert exactly to angles via $\cos\theta = 1 - D^2/2d$. θ_{pert} is the angle the perturbation turns the state through; θ_{nat} is the median angle between two clean examples at the same site. Where $\theta_{\text{pert}} > \theta_{\text{nat}}$ the perturbation is not a local displacement even after the radial component is set aside.

cell	edge	σ (=N/S)	$\|z_{\text{pert}}\|/\|z\|$	θ_{pert}	θ_{nat}	$\theta_{\text{pert}} - \theta_{\text{nat}}$
HRM/Maze	within-H	4.31	4.4×	76.9°	85.7°	-8.7°
HRM/Maze	within-L	6.22	6.3×	80.9°	38.7°	+42.2°
HRM/Maze	cross-HH	2.13	2.3×	64.8°	2.5°	+62.3°
TRM/Maze	within-H	15.28	15.3×	86.3°	62.1°	+24.2°
TRM/Maze	within-L	49.12	49.1×	88.8°	63.9°	+24.9°
TRM/Maze	cross-HH	25.31	25.3×	87.7°	5.2°	+82.5°
HRM/Sudoku	within-H	3.13	3.3×	72.3°	70.5°	+1.8°
HRM/Sudoku	within-L	12.63	12.7×	85.5°	54.6°	+30.9°
HRM/Sudoku	cross-HH	1.87	2.1×	61.9°	4.9°	+57.0°
TRM/Sudoku	within-H	7.72	7.8×	82.6°	75.7°	+6.9°
TRM/Sudoku	within-L	49.19	49.2×	88.8°	76.0°	+12.8°
TRM/Sudoku	cross-HH	4.27	4.4×	76.8°	11.9°	+64.9°
HRM/ARC	within-H	1.00	1.4×	45.0°	79.8°	-34.8°
HRM/ARC	within-L	1.00	1.4×	45.0°	79.4°	-34.4°
HRM/ARC	cross-HH	1.00	1.4×	45.0°	23.7°	+21.3°
TRM/ARC	within-H	1.00	1.4×	45.0°	81.6°	-36.6°
TRM/ARC	within-L	1.00	1.4×	45.0°	80.5°	-35.5°
TRM/ARC	cross-HH	1.00	1.4×	45.0°	41.7°	+3.3°

Table 29: Calibration audit: the self-drop actually achieved at the σ used, measured on $n = 200$ accepted examples with a percentile bootstrap over examples (2000 draws, seed 42) and 5 independent noise draws. σ was fitted by bisection on a *10-sample* estimate of the same quantity, against a target of 0.30, with an unseeded noise draw — so the fit could not resolve the target finely, and this is the first check of the result. 5 of 12 self-drop-calibrated analogs have a 95% CI that *excludes* the target, in both directions. The refit column re-runs the original 10-sample bisection: it is reported, not a gate, because the unseeded draw makes exact reproduction impossible by construction. ARC rows use `snr@1.0`, a closed form with no behavioural target; their achieved drop is reported but not scored.

cell	edge	σ	achieved	95% CI	refit σ	verdict
HRM/Maze	within-H	4.31	0.265	[0.205, 0.330]	6.26	consistent
HRM/Maze	within-L	6.22	0.230	[0.175, 0.285]	7.47	**under**
HRM/Maze	cross-HH	2.13	0.285	[0.225, 0.345]	2.22	consistent
TRM/Maze	within-H	15.28	0.505	[0.435, 0.575]	11.77	**over**
TRM/Maze	within-L	49.12	0.280	[0.215, 0.345]	49.96	consistent
TRM/Maze	cross-HH	25.31	0.360	[0.295, 0.430]	17.97	consistent
HRM/Sudoku	within-H	3.13	0.285	[0.225, 0.350]	3.01	consistent
HRM/Sudoku	within-L	12.63	0.615	[0.550, 0.680]	6.06	**over**
HRM/Sudoku	cross-HH	1.87	0.525	[0.460, 0.590]	1.62	**over**
TRM/Sudoku	within-H	7.72	0.315	[0.255, 0.375]	7.80	consistent
TRM/Sudoku	within-L	49.19	0.255	[0.195, 0.315]	43.47	consistent
TRM/Sudoku	cross-HH	4.27	0.395	[0.330, 0.465]	3.13	**over**
HRM/ARC	within-H	1.00	0.035	[0.010, 0.060]	2.85	*no target*
HRM/ARC	within-L	1.00	0.015	[0.000, 0.035]	4.06	*no target*
HRM/ARC	cross-HH	1.00	0.050	[0.025, 0.080]	2.21	*no target*
TRM/ARC	within-H	1.00	0.025	[0.005, 0.050]	2.75	*no target*
TRM/ARC	within-L	1.00	0.005	[0.000, 0.015]	2.26	*no target*
TRM/ARC	cross-HH	1.00	0.020	[0.005, 0.040]	14.29	*no target*

Table 30: Displacement across the six scale calibrations the robustness sweep already ran, at zero additional compute: the inject-site displacement is $\|\sigma\epsilon\| = \sigma\sqrt{d}$ by construction and the natural spread at that site is already measured, so the ratio follows for any σ. Read with the sweep's own ordering-preservation counts to see which calibrations are both near-manifold and order-preserving. Distribution and direction settings are excluded: they reuse the reference σ but not isotropic Gaussian noise.

cell	edge	setting	σ	ratio	radial	θ_{pert}
HRM/ARC	within-H	self_drop@10	1.35	1.05	1.7×	53.4°
HRM/ARC	within-H	self_drop@30	3.17	2.47	3.3×	72.5°
HRM/ARC	within-H	self_drop@50	2.26	1.76	2.5×	66.1°
HRM/ARC	within-H	snr@0.5	0.50	0.39	1.1×	26.6°
HRM/ARC	within-H	snr@1.0	1.00	0.78	1.4×	45.0°
HRM/ARC	within-H	snr@2.0	2.00	1.56	2.2×	63.4°
HRM/ARC	within-L	self_drop@10	2.76	2.16	2.9×	70.1°
HRM/ARC	within-L	self_drop@30	4.40	3.45	4.5×	77.2°
HRM/ARC	within-L	self_drop@50	4.64	3.63	4.7×	77.8°
HRM/ARC	within-L	snr@0.5	0.50	0.39	1.1×	26.6°
HRM/ARC	within-L	snr@1.0	1.00	0.78	1.4×	45.0°
HRM/ARC	within-L	snr@2.0	2.00	1.57	2.2×	63.4°
HRM/ARC	cross-HH	self_drop@10	0.93	2.26	1.4×	42.9°
HRM/ARC	cross-HH	self_drop@30	2.06	5.00	2.3×	64.1°
HRM/ARC	cross-HH	self_drop@50	3.33	8.10	3.5×	73.3°
HRM/ARC	cross-HH	snr@0.5	0.50	1.22	1.1×	26.6°
HRM/ARC	cross-HH	snr@1.0	1.00	2.43	1.4×	45.0°
HRM/ARC	cross-HH	snr@2.0	2.00	4.86	2.2×	63.4°
HRM/Maze	within-H	self_drop@10	3.52	2.59	3.7×	74.2°
HRM/Maze	within-H	self_drop@30	5.67	4.17	5.8×	80.0°
HRM/Maze	within-H	self_drop@50	6.71	4.94	6.8×	81.5°
HRM/Maze	within-H	snr@0.5	0.50	0.37	1.1×	26.6°
HRM/Maze	within-H	snr@1.0	1.00	0.74	1.4×	45.0°
HRM/Maze	within-H	snr@2.0	2.00	1.47	2.2×	63.4°
HRM/Maze	within-L	self_drop@10	3.54	5.34	3.7×	74.2°
HRM/Maze	within-L	self_drop@30	8.38	12.64	8.4×	83.2°
HRM/Maze	within-L	self_drop@50	10.58	15.96	10.6×	84.6°
HRM/Maze	within-L	snr@0.5	0.50	0.75	1.1×	26.6°
HRM/Maze	within-L	snr@1.0	1.00	1.51	1.4×	45.0°
HRM/Maze	within-L	snr@2.0	2.00	3.02	2.2×	63.4°
HRM/Maze	cross-HH	self_drop@10	1.84	41.42	2.1×	61.4°
HRM/Maze	cross-HH	self_drop@30	2.22	50.01	2.4×	65.7°
HRM/Maze	cross-HH	self_drop@50	2.68	60.52	2.9×	69.6°
HRM/Maze	cross-HH	snr@0.5	0.50	11.28	1.1×	26.6°
HRM/Maze	cross-HH	snr@1.0	1.00	22.56	1.4×	45.0°
HRM/Maze	cross-HH	snr@2.0	2.00	45.12	2.2×	63.4°
HRM/Sudoku	within-H	self_drop@10	1.95	1.69	2.2×	62.9°
HRM/Sudoku	within-H	self_drop@30	3.13	2.71	3.3×	72.3°
HRM/Sudoku	within-H	self_drop@50	3.13	2.71	3.3×	72.3°
HRM/Sudoku	within-H	snr@0.5	0.50	0.43	1.1×	26.6°
HRM/Sudoku	within-H	snr@1.0	1.00	0.87	1.4×	45.0°
HRM/Sudoku	within-H	snr@2.0	2.00	1.73	2.2×	63.4°
HRM/Sudoku	within-L	self_drop@10	2.68	2.92	2.9×	69.5°
HRM/Sudoku	within-L	self_drop@30	5.43	5.92	5.5×	79.6°
HRM/Sudoku	within-L	self_drop@50	9.07	9.88	9.1×	83.7°
HRM/Sudoku	within-L	snr@0.5	0.50	0.54	1.1×	26.6°
HRM/Sudoku	within-L	snr@1.0	1.00	1.09	1.4×	45.0°
HRM/Sudoku	within-L	snr@2.0	2.00	2.18	2.2×	63.4°
HRM/Sudoku	cross-HH	self_drop@10	1.15	13.48	1.5×	49.1°
HRM/Sudoku	cross-HH	self_drop@30	1.97	23.02	2.2×	63.1°

continued on next page

Table 30, continued

cell	edge	setting	σ	ratio	radial	θ_{pert}
HRM/Sudoku	cross-HH	self_drop@50	2.04	23.87	2.3×	63.9°
HRM/Sudoku	cross-HH	snr@0.5	0.50	5.84	1.1×	26.6°
HRM/Sudoku	cross-HH	snr@1.0	1.00	11.69	1.4×	45.0°
HRM/Sudoku	cross-HH	snr@2.0	2.00	23.38	2.2×	63.4°
TRM/ARC	within-H	self_drop@10	0.75	0.58	1.3×	37.0°
TRM/ARC	within-H	self_drop@30	2.37	1.81	2.6×	67.1°
TRM/ARC	within-H	self_drop@50	3.06	2.34	3.2×	71.9°
TRM/ARC	within-H	snr@0.5	0.50	0.38	1.1×	26.6°
TRM/ARC	within-H	snr@1.0	1.00	0.77	1.4×	45.0°
TRM/ARC	within-H	snr@2.0	2.00	1.53	2.2×	63.4°
TRM/ARC	within-L	self_drop@10	1.57	1.22	1.9×	57.5°
TRM/ARC	within-L	self_drop@30	2.15	1.67	2.4×	65.1°
TRM/ARC	within-L	self_drop@50	2.65	2.05	2.8×	69.3°
TRM/ARC	within-L	snr@0.5	0.50	0.39	1.1×	26.6°
TRM/ARC	within-L	snr@1.0	1.00	0.77	1.4×	45.0°
TRM/ARC	within-L	snr@2.0	2.00	1.55	2.2×	63.4°
TRM/ARC	cross-HH	self_drop@10	2.69	3.78	2.9×	69.6°
TRM/ARC	cross-HH	self_drop@30	16.41	23.06	16.4×	86.5°
TRM/ARC	cross-HH	self_drop@50	17.57	24.68	17.6×	86.7°
TRM/ARC	cross-HH	snr@0.5	0.50	0.70	1.1×	26.6°
TRM/ARC	cross-HH	snr@1.0	1.00	1.40	1.4×	45.0°
TRM/ARC	cross-HH	snr@2.0	2.00	2.81	2.2×	63.4°
TRM/Maze	within-H	self_drop@10	5.48	5.31	5.6×	79.7°
TRM/Maze	within-H	self_drop@30	12.06	11.70	12.1×	85.3°
TRM/Maze	within-H	self_drop@50	19.74	19.14	19.8×	87.1°
TRM/Maze	within-H	snr@0.5	0.50	0.48	1.1×	26.6°
TRM/Maze	within-H	snr@1.0	1.00	0.97	1.4×	45.0°
TRM/Maze	within-H	snr@2.0	2.00	1.94	2.2×	63.4°
TRM/Maze	within-L	self_drop@10	24.97	23.59	25.0×	87.7°
TRM/Maze	within-L	self_drop@30	49.12	46.42	49.1×	88.8°
TRM/Maze	within-L	self_drop@50	50.00	47.25	50.0×	88.9°
TRM/Maze	within-L	snr@0.5	0.50	0.47	1.1×	26.6°
TRM/Maze	within-L	snr@1.0	1.00	0.94	1.4×	45.0°
TRM/Maze	within-L	snr@2.0	2.00	1.89	2.2×	63.4°
TRM/Maze	cross-HH	self_drop@10	7.82	85.71	7.9×	82.7°
TRM/Maze	cross-HH	self_drop@30	26.80	293.63	26.8×	87.9°
TRM/Maze	cross-HH	self_drop@50	28.51	312.38	28.5×	88.0°
TRM/Maze	cross-HH	snr@0.5	0.50	5.48	1.1×	26.6°
TRM/Maze	cross-HH	snr@1.0	1.00	10.96	1.4×	45.0°
TRM/Maze	cross-HH	snr@2.0	2.00	21.91	2.2×	63.4°
TRM/Sudoku	within-H	self_drop@10	2.29	1.87	2.5×	66.4°
TRM/Sudoku	within-H	self_drop@30	8.20	6.68	8.3×	83.0°
TRM/Sudoku	within-H	self_drop@50	13.29	10.83	13.3×	85.7°
TRM/Sudoku	within-H	snr@0.5	0.50	0.41	1.1×	26.6°
TRM/Sudoku	within-H	snr@1.0	1.00	0.81	1.4×	45.0°
TRM/Sudoku	within-H	snr@2.0	2.00	1.63	2.2×	63.4°
TRM/Sudoku	within-L	self_drop@10	9.63	7.82	9.7×	84.1°
TRM/Sudoku	within-L	self_drop@30	49.19	39.95	49.2×	88.8°
TRM/Sudoku	within-L	self_drop@50	49.90	40.52	49.9×	88.9°
TRM/Sudoku	within-L	snr@0.5	0.50	0.41	1.1×	26.6°
TRM/Sudoku	within-L	snr@1.0	1.00	0.81	1.4×	45.0°
TRM/Sudoku	within-L	snr@2.0	2.00	1.62	2.2×	63.4°
TRM/Sudoku	cross-HH	self_drop@10	6.26	30.20	6.3×	80.9°
TRM/Sudoku	cross-HH	self_drop@30	12.51	60.35	12.5×	85.4°
TRM/Sudoku	cross-HH	self_drop@50	5.09	24.54	5.2×	78.9°
TRM/Sudoku	cross-HH	snr@0.5	0.50	2.41	1.1×	26.6°

continued on next page

Table 30, continued

cell	edge	setting	σ	ratio	radial	θ_{pert}
TRM/Sudoku	cross-HH	`snr@1.0`	1.00	4.82	1.4×	45.0°
TRM/Sudoku	cross-HH	`snr@2.0`	2.00	9.65	2.2×	63.4°

H.2 THE INPUT→OUTPUT EDGE IN EVERY CELL

The shared edge perturbs one cell's token embedding and reads the change in output logits at every cell, the same probe as the recursion control, at $n = 200$ with self-drop calibration per model (Table 31). TRM/Maze reproduces the recursion control's value (.200 against .189, null matching to four decimals). Both ARC-AGI rows sit at their null and should be read as absent structure rather than as results. The embedding site has no continuous activation distribution, since two cells holding the same token have identical embeddings, so the displacement of Appendix H.1 is undefined there; the well-posed comparison is against the token set's own geometry, and the injected vector exceeds the vocabulary's diameter in every cell (Table 32). The self-drop target is missed in four of six cells at this edge, all undershooting.

Table 31: Input→output locality at the embedding edge, for all six cells. The perturbation adds Gaussian noise to one cell's token embedding and the effect is read as the distribution of $\|\Delta\text{logits}\|$ over target cells (`embedding_patch`), so this is the same probe the recursion control uses and is architecture-agnostic. Before this, the paper reported this edge only for TRM/Maze and its non-recursive baseline. The geometric null is the mass the declared neighbourhood would receive under a uniform effect distribution; a row near its null carries little structure regardless of its absolute value.

cell	σ	locality [95% CI]	null	× null	n
HRM/Maze	0.081	0.1214 [0.1186, 0.1240]	0.0261	4.6	200
TRM/Maze	0.152	0.1998 [0.1934, 0.2064]	0.0261	7.7	200
HRM/Sudoku	0.034	0.2214 [0.2202, 0.2225]	0.1111	2.0	200
TRM/Sudoku	0.049	0.2094 [0.2068, 0.2120]	0.1111	1.9	200
HRM/ARC	0.101	0.4036 [0.3775, 0.4290]	0.3447	1.2	200
TRM/ARC	0.051	0.4673 [0.4422, 0.4921]	0.3995	1.2	200

Table 32: What the perturbation looks like at the embedding edge. $\|\sigma\epsilon\|$ is compared with the *token set's* own geometry rather than with a between-example spread: `embed_tokens` is a lookup table, so two cells holding the same token have identical embeddings and that spread is 0 for the majority of pairs (the *0-frac* column) — the clean distribution at this site is a finite set of at most $|V|$ points, with no continuous manifold to be on or off. *diam* is the largest distance between any two token embeddings in use. A ratio above 1 means the injected vector is larger than the entire vocabulary's extent, so the perturbed vector is not near any token. The achieved self-drop is measured at the σ used, as in Table 29.

cell	$\|\sigma\epsilon\|$	$\|emb\|$	N/S	0-frac	$\|\sigma\epsilon\|$/diam	achieved self-drop
HRM/Maze	1.83	0.362	5.1×	0.96	2.8×	0.130 [0.085, 0.180]
TRM/Maze	3.44	0.601	5.7×	0.96	2.8×	0.350 [0.285, 0.420]
HRM/Sudoku	0.78	0.231	3.4×	0.49	2.0×	0.090 [0.050, 0.130]
TRM/Sudoku	1.11	0.448	2.5×	0.48	1.9×	0.285 [0.225, 0.345]
HRM/ARC	2.28	0.678	3.4×	0.12	2.7×	0.170 [0.115, 0.225]
TRM/ARC	1.16	0.736	1.6×	0.14	1.6×	0.145 [0.100, 0.195]