SHORTCUT LOCK-IN: THE DYNAMICS OF LEARNING AND ESCAPING SPURIOUS FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks can improve at the core task while continuing to follow a spurious cue on conflicting inputs. When does further training allow useful task information to take control? We investigate this gap between capability and decision control through an evidence-transport account that separates the supply of corrective examples from the decision progress induced by their updates. A tractable two-route model shows how continued core learning during shortcut dominance leads to eventual escape under persistent conflict evidence, with distinct capture and escape timescales. During recovery, ViT-Tiny preserves coarse semantics while its sample geometry changes more than ResNet-18's. Early measurements also carry timing information: on four prospectively held-out ViT-Tiny runs, first-ten-update estimates predict escape onset with 13.33% median absolute percentage error. Visual and multimodal experiments reproduce capture and recovery, and a five-seed GQA study extends the analysis to natural image–question pairs. These results distinguish the acquisition of useful task information from the training dynamics that make it decisive.

1 INTRODUCTION

Reliable generalization requires models to use task-relevant features when correlations in their training data change. Yet neural networks can achieve high in-distribution accuracy through shortcuts that fail under distribution shift (Geirhos et al., 2020). Understanding whether training can overcome such dependence is essential for assessing the reliability of visual and multimodal systems beyond their training environments.

Robustness methods demonstrate that shortcut dependence can be corrected. Regularized group DRO improves worst-group generalization by changing the training objective (Sagawa et al., 2020a). Deep feature reweighting (DFR) recovers robust predictions by retraining only the final layer on balanced data (Kirichenko et al., 2023). Its success reveals a central tension: *task-relevant (core) features can already be present, yet a spurious cue still controls prediction*. This raises a question about continued training under the original objective: when does available core information begin to control decisions?

Learning-dynamics studies partly explain this tension. Simpler predictive features can dominate training (Shah et al., 2020), while task-relevant information may remain available despite a spurious readout (Izmailov et al., 2022; Hermann et al., 2024). Gradient starvation and controlled feature-learning analyses further characterize feature competition (Pezeshki et al., 2021; Qiu et al., 2024), and SPARE exploits early shortcut dependence for mitigation (Yang et al., 2024). Recovery after shortcut capture adds a temporal question: *how long does shortcut dependence persist, and why does recovery differ across architectures?*

We study *post-capture recovery*: whether, when, and how models move from shortcut-dependent to core-sensitive decisions during continued training. We track shortcut capture, escape onset at core–shortcut crossover, and successful escape under a sustained-recovery criterion. Figure 1 illustrates the central divergence: similar early shortcut reliance can precede semantic escape or persist throughout the observed training horizon. Our guiding idea is that recovery depends both on how often training supplies evidence against a shortcut and on how effectively updates turn that evidence into core-sensitive decisions.

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

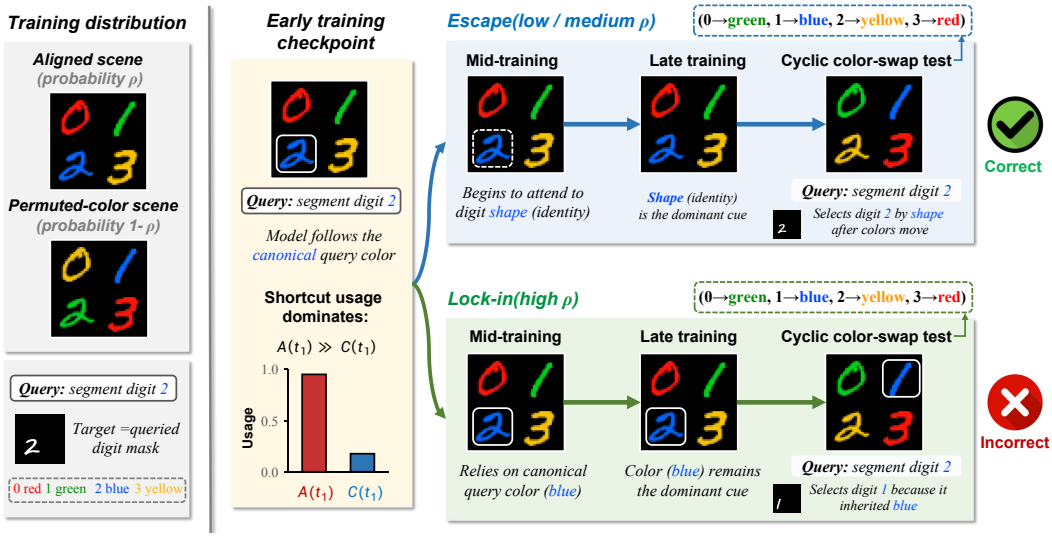


Figure 1: **Shortcut capture can precede semantic escape or finite-horizon lock-in.** Digit identity defines the queried segmentation mask, while the canonical digit–color association holds with probability ρ. After shortcut capture, training can yield identity-sensitive predictions or persistent color dependence. Under cyclic color swap, the queried digit 2 becomes yellow and digit 1 becomes blue: the core-following model selects 2 by identity, whereas the shortcut-following model selects 1 by its learned color cue. Lock-in is assessed within the observation horizon. The ρ labels indicate qualitative tendencies; outcomes also depend on architecture, seed, and training horizon. Selection boxes and usage bars are schematic.

Our contributions are threefold:

1. **We measure when core-task capability gains decision control.** Core-sensitive capability can improve before it governs predictions under conflict, and comparable shortcut capture can precede markedly different recovery times across architectures (Section 2).
2. **We develop an evidence-transport account of delayed shortcut escape.** The framework separates corrective-example supply from the decision progress induced by corrective updates, while a tractable two-route model yields distinct capture and escape timescales under persistent conflict evidence (Section 3).
3. **We characterize representation changes and early timing signals during recovery.** ViT-Tiny preserves coarse semantic structure while its sample geometry changes more than ResNet-18's, with attention contributing most to the measured reorganization. On four prospectively held-out ViT-Tiny runs, first-ten-update estimates predict escape onset with 13.33% median absolute percentage error. Additional experiments extend recovery across shortcut carriers and modalities (Sections 4–5).

2 Shortcut Capture Does Not Determine Recovery

Shortcut capture can precede either recovery or continued dependence within a finite training horizon. We first compare these trajectories, then ask whether delayed recovery reflects missing core-task capability.

2.1 Separating Acquisition from Decision Control

Our main task separates digit identity, which determines a queried segmentation mask, from color, which agrees with identity with probability $\rho \in \{0.80, 0.95, 0.995, 1.00\}$. Color-swap evaluation preserves the queried identity while changing the digit–color mapping. We compare U-Net (Ronneberger et al., 2015), ResNet-18 (He et al., 2016), and ViT-Tiny (Dosovitskiy et al., 2021) under common reliability conditions.

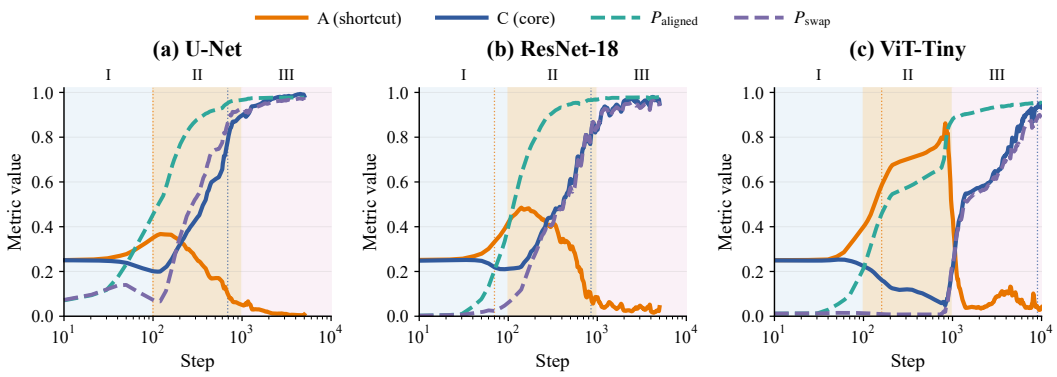


Figure 2: **Example post-capture trajectories differ across architectures.** At $\rho = 0.95$ (seed 3410), U-Net and ResNet-18 escape at steps 690 and 850, whereas ViT-Tiny escapes at step 9,000. Curves show shortcut-follow, core-follow, aligned performance, and counterfactual color-swap performance, with display-only EMA smoothing (span 11) and interpolation on a log-step grid. Dotted markers indicate focus and escape times from the raw records. Shaded bands I–III denote fixed step intervals (10–100, 100–1,000, and 1,000–10,000); event boundaries are marked separately.

Reading a recovery trajectory. Aligned and swapped performance are denoted by P_{align} and P_{swap}, while $A(t)$ and $C(t)$ measure shortcut-follow and core-follow behavior. Under the cyclic color swap, $A(t)$ measures allegiance to the digit carrying the queried digit's training color, whereas $C(t)$ measures allegiance to the queried digit itself. Their separation therefore tracks which cue controls the decision when shortcut and core information disagree. We distinguish shortcut capture (T_{focus}), escape onset at core–shortcut crossover (T_{cross}), and successful escape (T_{escape}) under a sustained-recovery criterion. Core capability can improve before crossover; unobserved events are right-censored. Exact event definitions and persistence rules appear in Appendix B.1.

The main reliability sweep contains 60 trajectories, with five seeds for every architecture–reliability condition at $\rho \in \{0.80, 0.95, 0.995, 1.00\}$. Appendix B.1 details these populations and the event conventions used in the prediction and extension studies. Increasing ρ jointly increases shortcut reliability and reduces conflict frequency.

2.2 Similar Capture, Different Recovery Times

With the task, reliability, and seed held fixed, all three architectures initially follow the shortcut, yet ViT-Tiny takes substantially longer to escape than U-Net and ResNet-18 (Figure 2). Similar early dependence can thus precede different recovery timescales. ViT-Tiny also fits aligned data more slowly; normalizing by time to 90% aligned performance reduces, but does not remove, the recovery-time gap (Appendix C.1).

Reliability changes the opportunity to recover. The wider sweep shows recovery slowing as shortcuts become more reliable (Table 1). At $\rho = 0.95$, all five U-Net and ResNet-18 runs escape within 5,000 steps, whereas none of the five ViT-Tiny runs does so within the same horizon. Extended ViT-Tiny trajectories reveal later recovery in some runs. At $\rho = 1$, where training contains no conflict examples, escape remains unobserved in every architecture. Escape is counted only among captured runs; core-first runs are reported separately.

The same conflict frequency therefore permits different durations of shortcut dependence across architectures. This distinction is not visible from aligned performance alone: models can fit the training distribution well while remaining at very different stages of shortcut recovery. The trajectory therefore separates ordinary task fitting from the later transfer of decision control to the core feature.

2.3 Core Availability Does Not Imply Decision Control

Shallow-MM tests whether core capability can become available while shortcut dependence persists. The answer is digit parity XOR a text operation; a separate text token agrees with it with probability

Table 1: **Recovery slows as shortcuts become more reliable.** Escape incidence and escape-time summaries are computed over the post-capture risk set; core-first runs are excluded (five independent seeds per condition).

		Shortcut capture	Escape		Final
Architecture	ρ	T_{focus} median [IQR]	Escaped / captured	T_{esc} KM median [95% CI]	Swap mean
U-Net	0.80	—†	1/1	550†	0.98
	0.95	70 [60, 100]	5/5	660 [540, 950]	0.95
	1.00	190 [180, 1 550]	0/5	NR	0.01
ResNet-18	0.80	—†	1/1	400†	0.97
	0.95	80 [80, 100]	5/5	530 [420, 850]	0.94
	1.00	210 [170, 400]	0/5	NR	0.01
ViT-Tiny	0.80	150 [100, 160]	3/5	4 800 [4 200, —]	0.89
	0.95	180 [170, 180]	2/5	11 850 [9 000, 11 850]	0.81
	1.00	190 [90, 200]	0/5	NR	0.01

Times are optimizer steps; escape is evaluated only over captured runs. † One captured run at $\rho = 0.80$; its observed T_{escape} is reported without an interval. NR: Kaplan–Meier median not reached; — in a CI: unidentified bound. Details: Appendices B.1 and C.1.

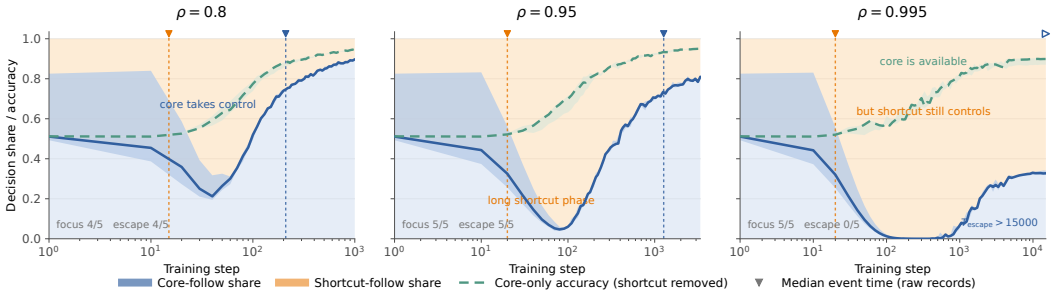


Figure 3: **Core availability and decision control in Shallow-MM.** Core-follow and shortcut-follow shares are evaluated under shortcut conflict; shortcut-removed accuracy measures performance without the shortcut token. Increasing ρ prolongs shortcut dominance; at $\rho = 0.995$, escape is right-censored beyond the recorded horizon.

ρ. At high reliability, shortcut-removed accuracy improves while predictions under conflict still follow the shortcut (Figure 3). Core capability can therefore be expressed before it controls decisions in the shortcut's presence. Section 5.2 examines recovery across the full cohort.

Finding 1. Core capability can precede decision control

In Shallow-MM, the model can solve the task without the shortcut yet still follow the shortcut when the cues disagree.

3 An Evidence-Transport Account of Recovery

The availability–control gap suggests separating two questions: how often training supplies evidence against a shortcut, and how much that evidence changes the model's decision. We call the latter effect *evidence transport*: signed progress in prediction space. We formalize this distinction under gradient flow and then study a tractable two-route model of delayed recovery.

3.1 Corrective Evidence Must Change the Decision

Let y_c be the core label and y_s the shortcut-indicated label under the swap distribution $\mathcal{Q}_{\text{sw}}$. The continuous decision gap is

$$D(\theta) = C(\theta) - A(\theta) = \mathbb{E}_{x \sim \mathcal{Q}_{\text{sw}}} \left[f_{y_c}(x;\theta) - f_{y_s}(x;\theta) \right], \tag{1}$$

where $C(\theta) = \mathbb{E}f_{y_c}$ and $A(\theta) = \mathbb{E}f_{y_s}$ are logit allegiances. These differentiable quantities are distinct from the empirical decision shares plotted in the experiments. Writing $L = \rho L_a + qL_c$, with $q = 1 - \rho$, gradient flow gives

$$\dot{D} = \rho\mathcal{T}_a + q\mathcal{T}_c, \qquad \mathcal{T}_j = -\langle \nabla D, \nabla L_j \rangle. \tag{2}$$

This chain-rule identity separates conflict frequency q from its signed effect $\mathcal{T}_c$. Conflict examples promote recovery when their gradients move the counterfactual decision toward the core label.

Architecture enters through the instantaneous output Jacobian $J_{\mathcal{A}} = \partial f/\partial\theta$ and kernel $K_{\mathcal{A}} = J_{\mathcal{A}}J_{\mathcal{A}}^{\top}$:

$$\mathcal{T}_j^{\mathcal{A}} = -g_D^{\top} K_{\mathcal{A}} r_j, \qquad g_D = \nabla_f D, \quad r_j = \nabla_f L_j. \tag{3}$$

The same conflict frequency can therefore yield different decision progress across trained networks. Event-aligned measurements reveal architecture-dependent directional opposition among corrective gradients and different signed transport at matched ResNet-18 and ViT-Tiny checkpoints (Appendix C.2.5).

Recovery as an accumulated evidence budget. At capture time $t_f = T_{\text{focus}}$, let $B_{\text{esc}} = b - D(t_f)$ be the remaining gap to a continuous threshold b. For a crossing $D(\tau_b) = b$,

$$B_{\text{esc}} = b - D(t_f) = \int_{t_f}^{\tau_b} [\rho\mathcal{T}_a(t) + q\mathcal{T}_c(t)]\, dt. \tag{4}$$

Defining the average signed rate over this interval as $\bar{v}$ yields

$$\tau_b - t_f = B_{\text{esc}}/\bar{v}. \tag{5}$$

A long recovery interval can reflect a large remaining gap, slow decision progress, or both. This decomposition separates two sources of delayed recovery: how far the model remains from core control at capture, and how effectively subsequent updates close that gap. Equation 5 is retrospective because its average rate is defined over the interval to the crossing; experimental event times additionally use checkpoint sampling and persistence criteria.

3.2 A Minimal Model Separates Capture from Escape

A two-route model specifies how the recovery rate evolves through competition between shortcut and core learning. Let U and V be shortcut and core product margins, with aligned and conflict margins $U + V$ and $V - U$. The loss is

$$L(U, V) = \rho\ell(U + V) + q\ell(V - U), \qquad \ell(m) = \log(1 + e^{-m}). \tag{6}$$

For balanced positive product routes of depths L_s, L_c and shortcut scale $\kappa > 0$, gradient flow reduces exactly to

$$\dot{U} = L_s\kappa^{2/L_s} U^{2-2/L_s} \Gamma_U(U, V), \tag{7}$$

$$\dot{V} = L_c V^{2-2/L_c} \Gamma_V(U, V), \tag{8}$$

where

$$\Gamma_U = \rho\sigma[-(U + V)] - q\sigma[-(V - U)], \tag{9}$$

$$\Gamma_V = \rho\sigma[-(U + V)] + q\sigma[-(V - U)]. \tag{10}$$

The model isolates a concrete mechanism: early shortcut amplification can outpace core learning, while the residual conflict signal later allows the core route to gain control.

For $1/2 < \rho < 1$ and positive initialization, $V(t)$ grows without bound, whereas $U(t) \leq \max\{U(0), U^{\star}\}$ with $U^{\star} = \log(\rho/q)$. Thus $V - U$ eventually crosses every finite threshold: lock-in at nonzero conflict rate is transient within this model.

Insight 1. Continued core learning enables delayed escape

In the two-route model, the core route keeps learning while the shortcut dominates, so lock-in at $\rho < 1$ is a finite-horizon effect.

For depth-two routes initialized by $U(0) = \kappa c_s \varepsilon^2$ and $V(0) = c_c \varepsilon^2$, let $a = \kappa(2\rho - 1)$. When $a > 1$ and the focus threshold is below $U^\star$,

$$T_{\text{focus}} = \frac{2}{a} \log(1/\varepsilon) + O(1), \tag{11}$$

$$T_{\text{escape}} - T_{\text{focus}} = \frac{1 - 1/a}{2\rho(1-\rho)} \log(1/\varepsilon) + O(1). \tag{12}$$

The shortcut-first regime considered here has $a > 1$. Capture is governed by early amplification, whereas the subsequent delay depends on residual conflict evidence and core-route mobility. The two timescales therefore arise from different parts of the learning process: shortcut amplification determines entry into shortcut dominance, whereas continued core growth determines when that dominance is overturned.

Independent numerical integration verifies the local rates, invariants, and asymptotic coefficients. Proofs and numerical checks appear in Appendices D and C.7 (Figure 12).

3.3 Connecting the Model to Network Measurements

The minimal model motivates three questions about network recovery: whether core structure persists, which parameter changes accompany representation reorganization, and how corrective updates differ across architectures. Self-attention updates can change both represented evidence and token routing. This provides a natural connection between representation reorganization and the accumulation of corrective updates. The representation measurements below ask whether coarse semantic structure survives this transition, while parameter-group rollback localizes which parts of the network change most strongly. Complementarily, event-aligned gradient measurements at $\rho = 0.95$ show greater directional opposition among ViT-Tiny corrective gradients than among those of the CNNs across five seeds (Appendix C.2.5). Together, these measurements connect the reduced recovery picture to observable changes in trained networks.

4 Representations and Timing during Recovery

We next examine architecture-dependent representation changes during recovery and whether measurements near the start of training anticipate escape onset.

4.1 Architecture-Dependent Representation Changes during Recovery

We align matched ResNet-18 and ViT-Tiny runs by normalized post-focus recovery position. ViT-Tiny preserves coarse semantics while its sample geometry changes more than ResNet-18's (Figure 4a,b). At matched intrinsic rank 16, the paired difference in held-out Procrustes NRMSE is 0.119 (95% CI $[0.072, 0.160]$). Temporal CKA supports this pattern, and sample-distance RSA is higher for ResNet-18 in every paired seed.

Masks retaining the top 25% of spatial evidence also preserve substantial output fidelity across architectures (Figure 4c). Thus *sample-geometry recomposition* accompanies recovery while coarse semantics and usable evidence persist.

Rolling selected ViT-Tiny parameter groups back to the preceding checkpoint further localizes the change: attention rollback produces the largest CKA restoration, exceeding MLP in all four seeds (Figure 9; Appendices B.2 and C.2). The combined picture is therefore not one of wholesale representation replacement. Coarse semantic organization remains recognizable while the geometry of individual examples is reorganized, especially through attention-associated updates.

Finding 2. Semantics can persist while representations change

ViT-Tiny retains broad semantic structure while changing how individual examples are represented, with the largest measured reorganization associated with attention.

4.2 Early Measurements Track Escape Onset

Can measurements from the first ten optimizer updates anticipate a much later escape onset? We compare early estimates with observed crossover times.

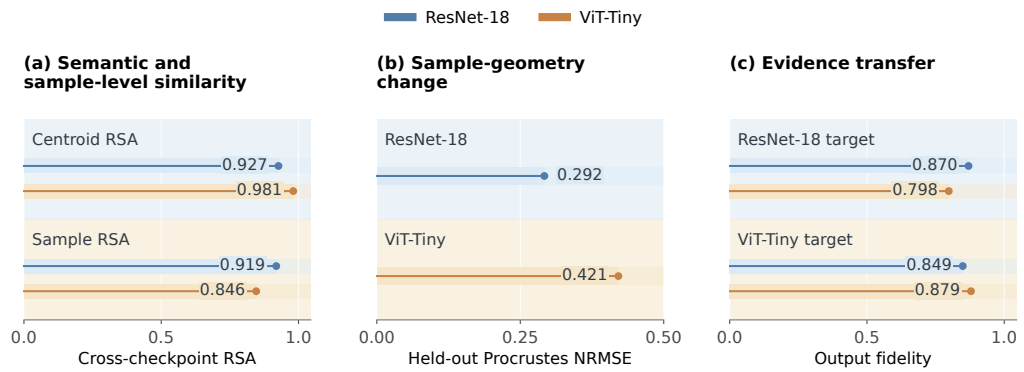


Figure 4: **Coarse semantics and usable evidence persist during sample-level reorganization.** (a) Centroid- and sample-distance RSA across matched post-focus recovery checkpoints. (b) Held-out Procrustes NRMSE at matched intrinsic rank 16; larger values indicate greater sample-geometry change. (c) Output fidelity after retaining the top 25% of spatial evidence, grouped by target; colors identify the mask source. Colors in (a,b) identify the architecture. Protocols and paired comparisons: Appendices B.2 and C.2.

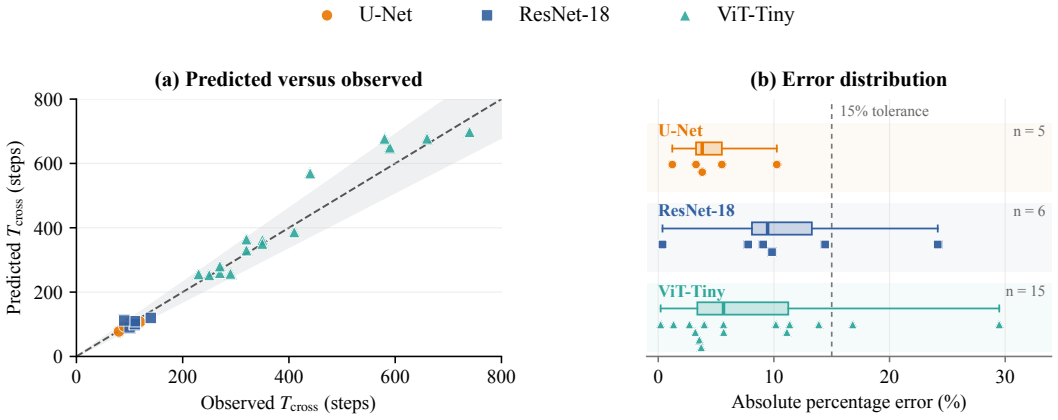


Figure 5: **Early training estimates track escape onset in the observed-crossover cohort.** Each marker represents one of the 26 runs with an observed crossover under the consecutive-evaluation criterion. Predictions use steps 0–10. (a) Predicted versus observed T_{cross}; the dashed diagonal indicates equality and the shaded band spans $\pm 15\%$ relative error. (b) Absolute percentage error distributions: boxes span the interquartile range, center lines mark medians, and whiskers show the full range. Points show all runs; vertical offsets separate overlapping points. The dashed line marks 15%.

The retrospective cohort contains 26 observed crossovers: five U-Net and six ResNet-18 runs at $\rho \in \{0.80, 0.90\}$, and fifteen ViT-Tiny runs at $\rho \in \{0.70, 0.80, 0.90\}$. Crossover in this cohort requires core-follow allegiance to exceed shortcut-follow allegiance across consecutive evaluations. Event conventions are detailed in Appendix B.6.

Crossovers occur at steps 80–740. Median absolute percentage error (APE) is 5.65%, with 23/26 predictions within 15% of the observed time (Figure 5). All three larger errors occur at $\rho = 0.90$. Appendix C.3 reports architecture summaries and per-run errors.

On four prospectively held-out ViT-Tiny runs at $\rho = 0.90$, the same early-timing analysis yields a median APE of 13.33%.

Finding 3. Early estimates track escape onset

Early trajectory measurements contain information about when core evidence later takes control.

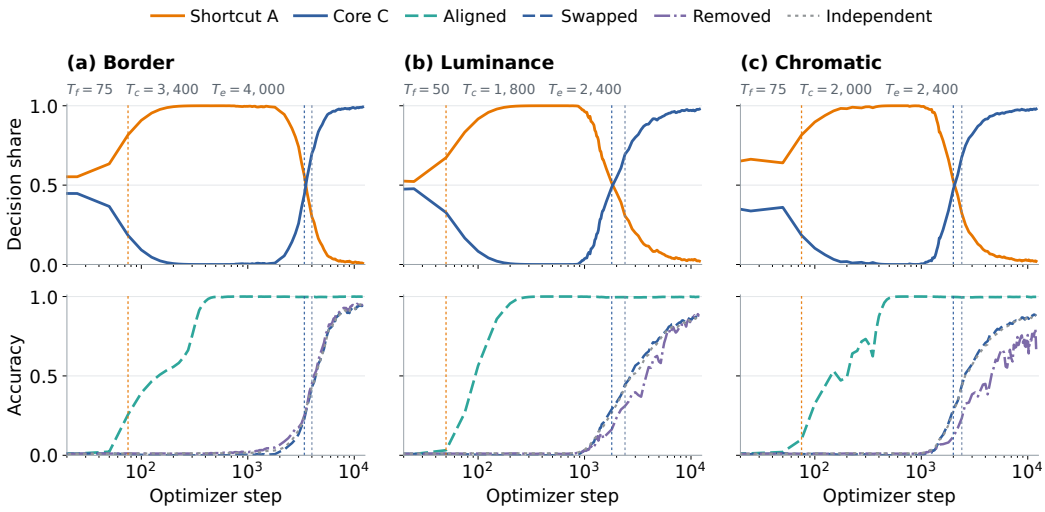


Figure 6: **Capture and recovery recur across three physical shortcut carriers.** Columns vary the rendering of a fixed Hadamard codebook, with $\rho = 0.90$, ViT-Tiny, optimizer, and seed 3407 held fixed. Top: shortcut and core decision shares on conflicting inputs. Bottom: accuracy under aligned, swapped, removed, and independent codes. Aligned performance rises before conflict performance in all three runs. Dotted lines mark recorded focus, crossover, and escape events (T_f, T_c, T_e); curves use an exponential moving average with span 3 for display. Each carrier is a single-seed run.

5 Scope of Shortcut Recovery

We test how broadly capture and recovery occur by varying the physical shortcut carrier, the modality, and fusion depth under natural language priors.

5.1 Recovery Persists across Physical Shortcut Carriers

To vary the physical carrier while controlling code similarity, we render a fixed equal-distance Hadamard codebook as a border stamp, a distributed luminance pattern, or an approximately isoluminant chromatic pattern. The core task is online classification of 100 jittered four-landmark constellations (Appendix B.3).

At fixed $\rho = 0.90$, ViT-Tiny, seed, and optimizer, all three runs first follow the shortcut and later recover constellation identity (Figure 6). Aligned accuracy improves before core decision share and conflict performance, with carrier-dependent recovery times and final performance. This shared ordering extends capture and recovery beyond a single physical cue. Each carrier uses one seed; full results and representation comparisons appear in Appendix C.4, with example inputs in Appendix B.3.

5.2 The Availability–Control Gap Extends across Modalities

The Shallow-MM task (Section 2.3) combines image and text in the core computation. Across five seeds per condition, captured runs of the 0.292M-parameter model escape at $\rho = 0.80$ and 0.95, whereas escape remains unobserved at $\rho = 0.995$ and 1.00 (Table 2).

Capture occurs in every run at both $\rho = 0.95$ and 0.995, yet recovery differs sharply. Together with the removal–conflict contrast in Figure 3, this shows that core capability and decision control can remain temporally separated across modalities. Appendix C.5 reports the full cohort and event times.

Table 2: **High shortcut reliability limits multimodal recovery.** Shallow-MM uses five seeds per reliability. Final text-swap accuracy is mean ± sample SD. Escape incidence is measured among captured runs; the $\rho = 0.80$ risk set excludes one core-first run.

Reliability ρ	Captured	Escaped after capture	Final text-swap
0.80	4/5	4/4	0.903 ± 0.010
0.95	5/5	5/5	0.801 ± 0.031
0.995	5/5	0/5	0.332 ± 0.017
1.00	5/5	0/5	0.000 ± 0.000

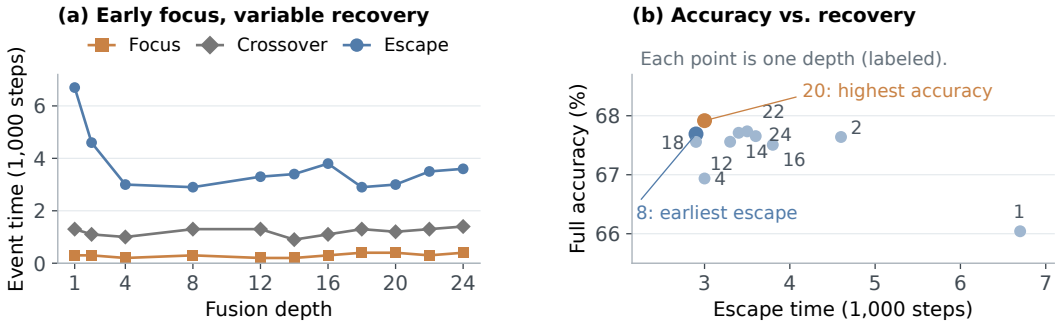


Figure 7: **Final accuracy conceals differences in GQA recovery time.** (a) Median focus, crossover, and successful-escape times across five independent seeds per fusion depth; lines are visual guides. (b) Mean full accuracy versus median escape time; labels indicate depth. All runs use $\rho = 0.75$; depth and parameter count vary jointly.

5.3 Final Accuracy Conceals Recovery-Time Differences in GQA

On a controlled 32-answer subset of GQA Balanced (Hudson & Manning, 2019), each question group's most frequent training answer defines a language-prior shortcut. A 3:1 aligned/conflict mixture fixes $\rho = 0.75$. A frozen DINOv2 ViT-S/14 encoder feeds a trainable fusion transformer across 11 depths from 1 to 24. Five independent seeds are run per depth, summarized in Figure 7. Successful escape requires $C - A \geq 0.20$ for three consecutive evaluations.

Across the five-seed depth sweep, mean final accuracy spans 66.0–67.9%, whereas median successful-escape times span 2,900–6,700 optimizer steps. The earliest median escape occurs at depths 8 and 18, while the highest mean final accuracy occurs at depth 20.

Insight 2. Recovery time complements final accuracy

Across five seeds, similar endpoint accuracy can coexist with different recovery-time distributions under the controlled language-prior shortcut.

First escape also differs from later retention: 8 of the 54 runs that escape finish below the escape margin, mostly at shallow depths (Appendices C.6.1 and C.6.2). Depth and parameter count vary jointly; Appendix A.4 gives a conditional scaling interpretation.

6 Discussion and Conclusion

Shortcut recovery concerns when available core evidence becomes decisive. The two-route model explains one source of prolonged dependence, while network experiments reveal representation changes and early timing signals that similar capture and final accuracy can conceal.

Limitations and future work. Transport measurements and interventions could test whether representation change controls recovery speed. Generalizing the timing results beyond observed crossovers requires held-out forecasting with fixed calibration, reliability-only baselines, and censored events. Parameter-matched GQA experiments could further separate fusion depth from model size. Capture, recovery, and retention together characterize shortcut dependence.

AI USE STATEMENT

We used generative AI tools to assist with literature retrieval and discovery, including identifying potentially relevant related work, as well as language editing and polishing of the manuscript. All AI-assisted work, including retrieved references and revised text, was reviewed and verified by the authors. The authors take responsibility for the final content of this work.

REFERENCES

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4971–4980, 2018.

Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing unimodal biases for visual question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Zheng-An Chen, Pengxiao Lin, Zhi-Qin John Xu, and Tao Luo. Focus and dilution: The multi-stage learning process of attention. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z. URL https://doi.org/10.1038/s42256-020-00257-z.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Katherine L. Hermann, Hossein Mobahi, Thomas Fel, and Michael C. Mozer. On the foundations of shortcut learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Tj3xLVuE9f.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew G. Wilson. On feature learning in the presence of spurious correlations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38516–38532. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/fb64a552feda3d981dbe43527a80a07e-Abstract-Conference.html.

Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. Roses are red, violets are blue... but should VQA expect them to? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2776–2785, 2021.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Zb6c8A-Fghk.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 3519–3529, 2019.

Tyler LaBonte and Vidya Muthukumar. SGD provably prioritizes a shortcut spurious feature in the XOR model. *arXiv preprint arXiv:2606.30444*, 2026. URL https://arxiv.org/abs/2606.30444.

Evan Z. Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 6781–6792, 2021.

Sibylle Marcotte, Gabriel Peyré, and Rémi Gribonval. Intrinsic training dynamics of deep neural networks. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IlyesljaNb.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. URL https://arxiv.org/abs/2304.07193.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1256–1272. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0987b8b338d6c90bbedd8631bc499221-Abstract.html.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022. URL https://arxiv.org/abs/2201.02177.

Guanwen Qiu, Da Kuang, and Surbhi Goel. Complexity matters: Feature learning in the presence of spurious correlations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41658–41697. PMLR, 2024. URL https://proceedings.mlr.press/v235/qiu24e.html.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 5301–5310, 2019.

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

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=ryxGuJrFvS.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 8346–8356, 2020b.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6120.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9573–9585. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6cfe0e6127fa25df2a0ef2ae1067d915-Abstract.html.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 10347–10357, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Yu Yang, Eric Gan, Gintare Karolina Dziugaite, and Baharan Mirzasoleiman. Identifying spurious biases early in training through the lens of simplicity bias. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2953–2961. PMLR, 2024. URL https://proceedings.mlr.press/v238/yang24c.html.

APPENDIX CONTENTS

A Technical Details 14

A.1 Related Work . . . 14
A.2 Class-Resolved Evidence Transfer . . . 15
A.3 Architecture and Temporal Transport Coherence . . . 15
A.4 From Transport to a Conditional Depth Law . . . 15

B Experimental Details 16

B.1 Task, Events, and Analysis Cohorts . . . 16
B.2 Representation Measurement Protocols . . . 17
B.3 Synthetic Multiple-Carrier Task . . . 18
B.4 Shallow Multimodal Task . . . 19
B.5 GQA Fusion-Depth Experiment . . . 19
B.6 Training-Start Crossover Prediction Protocol . . . 20

C Additional Experimental Results 20

C.1 Full Reliability and Robustness Results . . . 20
C.2 Post-Focus Representation Dynamics . . . 22
C.3 Training-Start Crossover Prediction . . . 23
C.4 Complete Carrier Results . . . 24
C.5 Complete Shallow Multimodal Cohort . . . 25
C.6 GQA Depth Sweep and Recovery Trajectories . . . 25
C.7 Numerical Verification . . . 27

D Proofs 28

D.1 Architecture-Conditioned Transport . . . 28
D.2 From Trajectory Approximation to Crossing-Time Error . . . 28
D.3 Balanced Product Routes . . . 29
D.4 Small-Initialization Asymptotics . . . 29

A Technical Details

A.1 Related Work

Shortcut reliance and robust readouts. Shortcut learning connects failures under distribution shift to predictive rules that exploit training-specific regularities (Geirhos et al., 2020). Robustness interventions address this dependence through different mechanisms: regularized group DRO optimizes worst-group performance (Sagawa et al., 2020a), whereas deep feature reweighting (DFR) retrains the final layer on balanced data (Kirichenko et al., 2023). DFR demonstrates that useful core features can coexist with shortcut-dependent predictions. These interventions establish that robust readouts are accessible; the timing of recovery under continued training with the original objective is a different question. We examine this temporal gap, distinguishing feature availability from decision control and separating capture, escape onset, and successful escape while retaining unobserved events as right-censored. Overparameterization can exacerbate reliance on spurious correlations (Sagawa et al., 2020b), while JTT improves group robustness by upweighting examples misclassified by an initial model (Liu et al., 2021). These works address capacity effects and training interventions; our measurements concern the trajectory of recovery under continued training.

Feature competition and early diagnostics. Gradient starvation formalizes how learning one predictive feature can suppress the acquisition of others (Pezeshki et al., 2021). Controlled studies further show that relative feature complexity and correlation strength affect core learning, with potentially overlapping learning phases (Qiu et al., 2024); recent XOR theory establishes shortcut-first learning and subsequent signal suppression for two-layer ReLU networks (LaBonte & Muthukumar, 2026). Early diagnostics also support intervention: SPARE identifies spurious biases from early outputs and uses importance sampling to mitigate them (Yang et al., 2024). These studies explain how shortcuts become dominant and how early signals guide mitigation. We extend the temporal analysis to recovery after capture, comparing its timing across architectures, characterizing representation changes, and distinguishing core-task capability from control under conflict. More broadly, spectral bias shows that different target components can be learned at different rates (Rahaman et al., 2019). Such learning-order effects provide context for feature competition, but do not by themselves determine when a core feature gains decision control under shortcut conflict.

Reduced dynamics and transition times. Exact analyses of deep linear networks connect learning plateaus and rapid transitions to tractable dynamical systems (Saxe et al., 2014). Marcotte et al. (2026) formalize conditions under which parameter-space gradient flow admits intrinsic dynamics in architecture-dependent lifted coordinates. Behavioral shortcut/core observables pose a distinct closure problem. We derive a full-network transport identity that separates conflict frequency from signed progress, and an exactly reducible two-route model with an eventual-escape guarantee and leading-order timing analysis. Chen et al. (2026) derive staged focus–dilution dynamics for a one-layer Transformer trained on Markovian data. Our balanced product-route model offers a different, more restricted account of shortcut capture and recovery; it is not derived from the full Transformer updates. Both analyses distinguish stages of learning, but the state variables, data assumptions, and mechanisms differ. Under explicit approximation and crossing assumptions, a separate bound relates trajectory error to timing error. Delayed transitions after early fitting are also reminiscent of grokking (Power et al., 2022); our setting differs in tracking a switch in feature control under persistent spurious correlation rather than delayed train-to-test generalization.

Representations and attention architectures. Transformers use attention to aggregate information (Vaswani et al., 2017); ViT and data-efficient image transformers extend this architecture to visual recognition (Dosovitskiy et al., 2021; Touvron et al., 2021). Representation comparisons have identified differences between vision transformers and convolutional networks (Raghu et al., 2021). SVCCA and CKA provide complementary approaches to comparing learned representations (Raghu et al., 2017; Kornblith et al., 2019). We use temporal CKA and sample-geometry measurements to characterize changes during recovery, without equating representation similarity with causal feature use.

Language priors in visual question answering. VQA-CP exposes dependence on changing answer priors (Agrawal et al., 2018), and RUBi reduces unimodal biases during training (Cadene et al., 2019). GQA-OOD studies generalization under answer-distribution shifts (Kervadec et al., 2021). Our GQA experiment instead uses an explicitly selected and resampled subset to track recovery over

training. It should not be interpreted as a result on the standard VQA-CP or GQA-OOD evaluation protocol.

A.2 Class-Resolved Evidence Transfer

For query class i, define a gap D_i and decompose the conflict loss by source class j with prior π_j. The chain-rule argument gives

$$\dot{D}_i = \rho \mathcal{T}_{a,i} + (1-\rho) \sum_j \pi_j M_{ij}, \qquad M_{ij} = -\langle \nabla D_i, \nabla L_{c,j} \rangle. \tag{13}$$

Off-diagonal entries quantify how a core update from class j changes the counterfactual decision for class i. Their row sums are the classwise signed transport rates. Measurements of this operator along training would connect architecture-conditioned updates to classwise escape and allow comparisons of transport rates across architectures.

A.3 Architecture and Temporal Transport Coherence

The measurements in Section 4 distinguish representation stationarity from semantic organization. A first-order comparison describes how the representation carrier can enter the transport framework. For a residual convolutional block $h^+ = h + F(h; W)$,

$$\delta h^+ = (I + J_h F)\delta h + J_W F \, \delta W + O(\|\delta h\|^2 + \|\delta W\|^2). \tag{14}$$

Convolutional weight sharing provides a spatially tied coordinate system through which successive updates are propagated. The local Jacobian evolves during optimization, with the same parameterized filters reused across spatial locations. This shared structure is consistent with the comparatively stationary representations measured in Section 4. Alignment of successive corrective updates additionally depends on the local Jacobians and residuals.

For self-attention, $H^+ = H + P(H; Q, K)V(H)W_O$ with $P = \mathrm{softmax}(S)$ and $S = HQK^\top H^\top / \sqrt{d}$. Its perturbation includes

$$\delta P = J_{\mathrm{sm}}(S)[\delta S], \tag{15}$$

$$\delta S = \frac{(\delta H)QK^\top H^\top + H\,\delta(QK^\top)H^\top + HQK^\top(\delta H)^\top}{\sqrt{d}}. \tag{16}$$

In self-attention, the routing operator is itself sample-conditioned. The term involving $\delta(QK^\top)$ changes token-to-token interaction as a bilinear function of the current representation. Parameter updates can therefore modify both the represented evidence and the route through which that evidence is delivered to downstream readout.

These expressions suggest that changes in routing may affect how successive corrective updates accumulate.

For post-capture checkpoints $k = 1, \ldots, H$, write the conflict-induced functional update as $u_k = -K_{\mathcal{A},k} r_{c,k}$ and, for nonzero total update magnitude, define transport coherence

$$\mathcal{C}_H = \frac{\| \sum_{k=1}^H u_k \|}{\sum_{k=1}^H \|u_k\|} \in [0, 1]. \tag{17}$$

$\mathcal{C}_H$ distinguishes instantaneous corrective signal from its cumulative effect. When successive conflict-induced updates remain aligned, their functional displacements reinforce one another and $\mathcal{C}_H$ approaches one, with perfect alignment giving $\mathcal{C}_H = 1$. Rotation or sign reversal reduces the net displacement relative to the total transported magnitude. Routing changes could therefore slow progress through directional cancellation. Measurements of $\mathcal{C}_H$ and interventions on escape time would test this explanation.

A.4 From Transport to a Conditional Depth Law

Conditional transport interpretation. Let d denote fusion depth, $t_f(d)$ its focus time, and $\tau_b(d)$ the continuous logit-gap crossing. Equation 5 gives

$$\tau_b(d) - t_f(d) = \frac{B_b(d)}{\bar{v}(d)}, \qquad B_b(d) = b - D(t_f(d); d). \tag{18}$$

At fixed ρ, conflict frequency is controlled while $B_b(d)$ and $\bar{v}(d)$ can vary with depth. The instantaneous kernel also depends on the current parameters and input distribution.

A saturation law follows if $t_f(d) \approx t_f$ and $B_b(d) \approx B > 0$, and useful transport has the finite-depth form $\bar{v}(d) = v_\infty - cd^{-\beta}$ with $v_\infty, c, \beta > 0$ and $cd^{-\beta} < v_\infty$:

$$\begin{aligned} \tau_b(d) &\approx t_f + \frac{B}{v_\infty - cd^{-\beta}} \\ &= T_\infty + K_d d^{-\beta} + O(d^{-2\beta}), \\ T_\infty &= t_f + \frac{B}{v_\infty}, \qquad K_d = \frac{Bc}{v_\infty^2}. \end{aligned} \tag{19}$$

Here T_∞ is the limiting time within the assumed model and K_d is the finite-depth time penalty. The saturation form and exponent follow from the assumed depth dependence of transport. The nonmonotonic sweep in Section 5.3 provides a descriptive comparison with this law; direct transport measurements would test its rate assumption. Fusion depth d is distinct from the multiplicative route depths L_s, L_c in Equation 8.

Equation 19 follows by expanding $(1-x)^{-1} = 1 + x + O(x^2)$ for $x = c/(v_\infty d^\beta) \ll 1$. It requires an approximately depth-independent focus time and escape budget, a positive limiting transport rate, and the assumed finite-depth rate deficit. The median focus times in Table C7 are early (200–400 steps) relative to median escape (2,900–6,700 steps). Their variation and the unmeasured budget $B_b(d)$ affect this approximation.

The depth sweep provides a descriptive comparison. A free-exponent fit of Equation 19 to the five-seed median escape times (Equation 27) does not identify β: removing depth 1 or depth 2 moves the estimate to the boundary of the search range, and a bootstrap over seeds gives a 95% interval of $[1.0, 6.0]$. The sweep shows a shallow-depth penalty followed by nonmonotonic escape times at larger depths.

The exact transport equations concern a differentiable logit gap under gradient flow. The GQA events use empirical decision shares under discrete AdamW updates with warmup, decay, and weight decay. Applying the transport interpretation to these events involves assumptions linking the two observables and the two optimization settings. Each depth has five seeds, and parameter count increases with depth. Parameter-matched controls, and direct budget and transport measurements would distinguish these effects and test the scaling exponent.

B Experimental Details

B.1 Task, Events, and Analysis Cohorts

Each example contains a task-relevant digit shape and a shortcut color. In the main reliability sweep, $\rho \in \{0.80, 0.95, 0.995, 1.00\}$ is the probability that color agrees with the target. Shape remains label-determining on conflict examples. Color-swap evaluation changes the digit–color mapping while preserving the queried identity. U-Net, ResNet-18, and ViT-Tiny are compared on this task using common reliability conditions and evaluation observables. The additional prediction and extension cohorts use the reliability values specified in their respective sections.

Observables and event conventions. Let $P_{\text{align}}(t)$ and $P_{\text{swap}}(t)$ denote aligned and shortcut-swapped performance, and $A(t)$ and $C(t)$ the empirical shortcut- and core-follow metrics: under the cyclic color swap, the shares of predicted foreground mass on the digit carrying the queried digit's training color and on the queried digit. Shortcut dominance begins at the first of two consecutive evaluations with $A - C \geq 0.10$. Escape onset T_{cross} is the first of three consecutive evaluations after dominance with $C - A \geq 0$, the same crossover rule used for GQA; shortcut capture T_{focus} is the first local maximum of $A(t)$ after dominance and before T_{cross}; and successful escape T_{escape} is the first of three consecutive evaluations with $C \geq 0.80$, $A \leq 0.20$, and $P_{\text{swap}} \geq 0.90$. All events are recomputed from the saved evaluation trajectories. T_{robust} records the additional performance milestone of robust entry: the first of three consecutive evaluations with color-swap performance $P_{\text{swap}} \geq 0.90$. The escape records use three consecutive qualifying evaluations. Lock-in denotes persistent shortcut dependence within the observation horizon. Events are detected per run before aggregation; unobserved events are right-censored at each run's observation horizon.

The training-start prediction cohort uses the same crossover rule (Section 4.2; Appendix B.6). GQA specifies a behavioral-gap escape threshold in Section 5.3. The tasks share the capture–onset–escape interpretation, with task-specific observables and thresholds. These empirical events are distinct from the continuous logit-gap crossing used in the theory. We use *post-capture* for recovery following an observed focus event. The representation analysis uses post-focus recovery alignment (Appendix B.2); the synthetic and GQA summaries explicitly report $T_{\text{escape}} - T_{\text{cross}}$ where that narrower interval is used. The I–III bands in Figure 2 mark fixed display intervals; dotted lines locate each run's events.

Analysis cohorts. The main sweep contains five seeds for each architecture–reliability condition. Capture incidence is reported over all runs, whereas post-capture escape is evaluated only among runs that first enter shortcut dominance. Core-first runs are therefore kept separate from recovery after capture. Representation comparisons use matched shortcut-to-core cohorts.

Observation horizons. ViT-Tiny runs at $\rho = 0.95$ were initially trained for 5,000 steps. Runs whose P_{swap} continued to increase over the final 500 steps were extended to 10,000 steps, and runs still increasing over the final 1,000 steps were extended to 20,000 steps. Extension therefore depended on the observed P_{swap} trend rather than on whether the escape event had occurred. Under the common 5,000-step horizon, ViT-Tiny escapes in 0/5 runs at $\rho = 0.95$, compared with 5/5 for U-Net and ResNet-18. Because extension depended on recovery-related behavior, Kaplan–Meier timing summaries for these runs are descriptive; the common-horizon incidence provides the direct architecture comparison.

Table estimators. Table 1 reports capture times as the median and interquartile range (IQR). For the two conditions with only one observed capture, the median and IQR are omitted and the four uncaptured runs are classified as core-first. Escape counts and times are restricted to captured runs, using each run's horizon; core-first runs are excluded. Escape times are Kaplan–Meier medians, with 95% confidence intervals obtained by inverting Greenwood log–log confidence bands; with a single captured run, its observed escape time is reported without an interval. NR denotes an unreached median; a dash inside a confidence interval denotes an unidentified bound. Final swap performance is the across-seed mean rounded to two decimals. Appendix C.1 lists the full per-seed events, censoring indicators, and trajectory classes.

Varying ρ measures the joint effect of shortcut reliability and conflict frequency. The no-conflict endpoint $\rho = 1$ is a boundary condition, and lock-in is defined over the observation horizon.

B.2 Representation Measurement Protocols

Checkpoint alignment. The analysis uses matched shortcut-to-core runs and checkpoints from the ResNet-18 and ViT-Tiny post-focus recovery cohorts. Each recovery interval is mapped to normalized position

$$q(t) = \frac{t - T_{\text{focus}}}{T_{\text{rec,end}} - T_{\text{focus}}} \in [0, 1],$$

and cross-architecture comparisons pair checkpoints by seed and normalized q. This alignment separates representation-path geometry from differences in absolute recovery duration.

Sparse evidence transfer. The sparse output-fidelity protocol constructs an evidence mask in a source architecture, retains the top 25% of the evidence scores, and measures output fidelity after transferring the retained spatial evidence to a target architecture.

Temporal stability. Temporal CKA is computed between frozen representations at matched normalized post-focus recovery checkpoints using a common evaluation sample and centering convention.

Sample and semantic geometry. The geometry analysis uses 48 frozen checkpoints, 2,320 temporal-geometry rows, and 40 dimension-matched controls. Representations are matched to intrinsic rank 16 before cross-checkpoint geometry is evaluated. An orthogonal Procrustes map is fit without the evaluation samples and applied to the held-out representation. For held-out matrices X and Y and fitted orthogonal map $\widehat{R}$, the normalized error is

$$\text{NRMSE}_{\text{Proc}} = \frac{\|X\widehat{R} - Y\|_F}{\|Y\|_F}. \tag{20}$$

Sample-distance RSA is the correlation between vectorized held-out pairwise-sample distance matrices from two checkpoints. Centroid-distance RSA applies the same construction to the four-class centroid distance matrix. The final query-conditioned spatial attention pattern is compared between adjacent checkpoints by cosine similarity.

Selected parameter-group rollback. For two adjacent frozen ViT-Tiny checkpoints, the newer checkpoint supplies all parameters except one selected group, which is replaced by its value at the preceding checkpoint. Recovery is the increase in CKA similarity to the preceding representation relative to the unmodified newer checkpoint. The analysis contains 200 parameter-group rollbacks and 240 block-by-q localization rows.

B.3 Synthetic Multiple-Carrier Task

The synthetic task changes the physical carrier while holding the codebook, reliability, core task, optimizer, and model fixed.

Definition 1 (Core constellation distribution)**.** Let $K = 100$ and $Y \sim \mathrm{Unif}\{0, \dots, K-1\}$. Class y is assigned a deterministic prototype of four landmarks on a 7×7 grid, using a circle, diamond, cross, and square. On each online draw, positions are jittered independently,

$$\widetilde{p}_{yr} = \mathrm{clip}(p_{yr} + \xi_r), \qquad \xi_r \sim \mathcal{N}(0, \sigma_p^2 I), \quad \sigma_p = 4, \tag{21}$$

then rendered in a 64×64 RGB image with sampled radius and intensity. The background is Gaussian with mean 0.08 and standard deviation 0.025. The label is the constellation identity Y.

Definition 2 (Balanced Hadamard shortcut)**.** Let $H_{128} \in \{-1, +1\}^{128 \times 128}$ be a Sylvester Hadamard matrix and select 100 balanced nonconstant rows h_y. The shortcut label Z satisfies

$$\Pr(Z = Y \mid Y) = \rho, \qquad \Pr(Z = z \neq Y \mid Y) = \frac{1-\rho}{K-1}. \tag{22}$$

Orthogonality gives $\langle h_i, h_j \rangle = 0$ and $d_H(h_i, h_j) = 64$ for $i \neq j$. Each class therefore has the same pairwise shortcut-code distances. The experiment uses $\rho = 0.90$ without bit flips.

Let X^{core} be the constellation and $\alpha = 0.35$. The common code is rendered in three ways. The *border* carrier reshapes 128 bits into a 4×32 stamp $B(h_Z)$ and writes

$$X_{0:4,0:32,:} = \tfrac{1}{2} + \alpha B(h_Z). \tag{23}$$

For the *luminance* carrier, a deterministic tiled map $M(h_Z) \in \{-1, +1\}^{64 \times 64}$ is added equally to all channels:

$$X = \mathrm{clip}(X^{\mathrm{core}} + \alpha M(h_Z)(1, 1, 1)). \tag{24}$$

For the approximately isoluminant *chromatic* carrier,

$$X = \mathrm{clip}\left(X^{\mathrm{core}} + \alpha M(h_Z)\left(1, -\frac{0.299}{0.587}, 0\right)\right), \tag{25}$$

which cancels under standard linear luminance weights before clipping.

Evaluation uses aligned ($Z = Y$), cyclic swap ($Z = (Y+1) \bmod K$), removed, and independent-shortcut distributions. On the cyclic swap let $p_c = \Pr(\widehat{Y} = Y)$ and $p_s = \Pr(\widehat{Y} = Z)$, with decision shares $C = p_c/(p_c+p_s)$ and $A = p_s/(p_c+p_s)$. Focus uses the base-task shortcut-capture definition, while crossover and escape use the corresponding three-consecutive-evaluation persistence rules.

The corresponding carrier trajectories appear in Figure 6 in the main text.

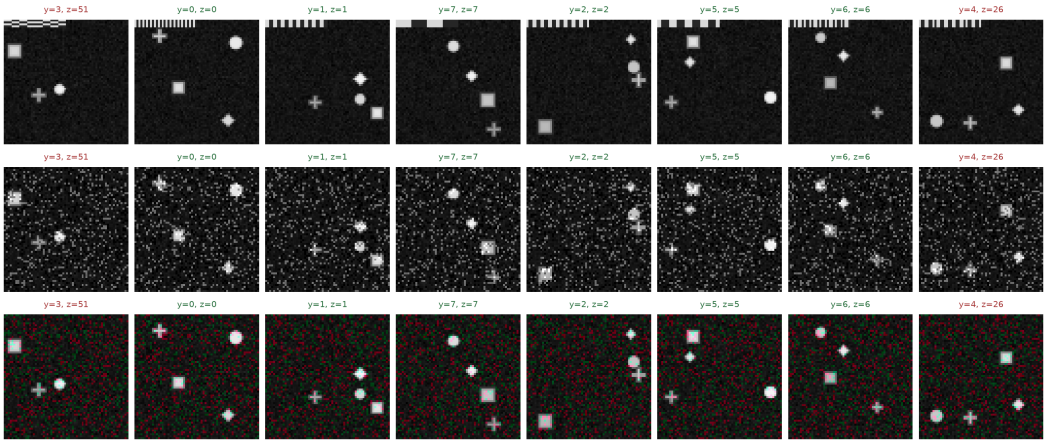


Figure 8: **Online synthetic constellation samples.** Rows show border, luminance, and chromatic renderings of the same equal-distance Hadamard codebook. Green annotations are aligned; red annotations are conflicting.

B.4 Shallow Multimodal Task

The Shallow-MM task pairs digit parity with a text operation through XOR; a separate text token agrees with the answer with probability ρ. The 0.292M-parameter model is evaluated across five seeds per condition.

B.5 GQA Fusion-Depth Experiment

B.5.1 Data and Natural Shortcut Construction

The experiment uses GQA Balanced (Hudson & Manning, 2019), restricted to single-object questions about color, material, category, and simple state. Target object regions are obtained from GQA scene graphs, enlarged with 10% context, and resized to 224×224. The answer vocabulary contains the 32 most frequent answers in the training set. For each question group g, define the shortcut answer from training frequencies,

$$y_s(g) = \arg\max_y n_{\text{train}}(g, y). \tag{26}$$

Within each group, prior-aligned ($y = y_s(g)$) and prior-conflicting ($y \neq y_s(g)$) examples are sampled at a 3:1 ratio, yielding $\rho = 0.75$. The shortcut arises from natural language priors in the selected data. Image, question, and answer content are preserved apart from the stated region preprocessing and sample selection. All depths use the same data splits and diagnostic examples, with five independent seeds per depth. Event-time summaries aggregate the five runs at each depth, and final accuracy is reported as the across-seed mean and sample standard deviation.

B.5.2 Architecture, Optimization, and Diagnostics

The visual encoder is a frozen DINOv2 ViT-S/14 (Oquab et al., 2023). Text embeddings are randomly initialized. A trainable multimodal Fusion Transformer uses hidden dimension 256, four attention heads, MLP ratio 4, and dropout 0.1; a `[CLS]` representation predicts one of 32 answers. Only fusion depth is varied as an architectural hyperparameter, with the approximate total parameter counts given in Table C7. Those counts include both the frozen visual encoder and trainable parameters.

Training uses AdamW, batch size 64, peak learning rate 3×10^{-4}, 500 warmup steps, cosine decay, weight decay 0.01, and a maximum of 20,000 optimizer steps. Behavioral trajectories are recorded every 100 steps. Question-only and image-shuffle accuracy are evaluated every 500 steps, alongside the tracked shortcut-follow $A(t)$, core-follow $C(t)$, decision gap $D_{\text{beh}}(t) = C(t) - A(t)$, and full accuracy. The analysis uses full accuracy, behavioral gaps, and event times (Table C7).

Event convention. Behavioral events are detected directly from the saved 100-step evaluation trajectories. Let $D_{\mathrm{beh}}(t) = C(t) - A(t)$. Crossover is the first of three consecutive evaluations with $D_{\mathrm{beh}} \geq 0$, and successful escape is the first of three consecutive evaluations with $D_{\mathrm{beh}} \geq 0.20$,

$$T_{\mathrm{escape}} = \min\{t_k : D_{\mathrm{beh}}(t_{k+j}) \geq 0.20 \text{ for } j = 0, 1, 2\}. \tag{27}$$

The event time is the first checkpoint in the qualifying sequence. T_{focus} retains the experiment's shortcut-capture annotation. Unobserved crossover or escape events are right-censored at the 20,000-step horizon.

Two distinct intervals. We denote the crossover-to-escape interval by $\Delta_{\mathrm{cross}\to\mathrm{escape}} = T_{\mathrm{escape}} - T_{\mathrm{cross}}$. The theoretical post-focus interval is $\Delta_{\mathrm{post}} = T_{\mathrm{escape}} - T_{\mathrm{focus}}$, so that

$$\Delta_{\mathrm{post}} = (T_{\mathrm{cross}} - T_{\mathrm{focus}}) + \Delta_{\mathrm{cross}\to\mathrm{escape}}. \tag{28}$$

The post-focus interval includes the time from capture to crossover as well as the crossover-to-escape interval.

B.6 Training-Start Crossover Prediction Protocol

The prediction dataset contains 26 observed and predicted crossover-time pairs, with architecture, reliability, seed, and absolute percentage error. Inputs cover steps 0–10: initialization and the first ten optimizer updates. Crossover requires core-follow allegiance to exceed shortcut-follow allegiance across consecutive evaluations. We evaluate timing agreement within this observed-crossover cohort.

For run i, we compute absolute percentage error as

$$\mathrm{APE}_i = \frac{|\widehat{T}_{\mathrm{cross},i} - T_{\mathrm{cross},i}|}{T_{\mathrm{cross},i}}. \tag{29}$$

Absolute percentage errors are computed from unrounded observed and predicted times. Pooled statistics weight each run equally, so architectures contribute in proportion to their cohort sizes. Table C4 lists every run; numerical results appear in Appendix C.3. Predictions are rounded to two decimal places for display.

Event labels.

Observed crossover times follow the crossover rule of Appendix B.1: the first of three consecutive evaluations after shortcut dominance with $C - A \geq 0$. Recomputing this rule from the saved trajectories reproduces the observed T_{cross} of all 22 runs whose trajectories are archived with the main records; the four ViT-Tiny runs at $\rho = 0.90$, seeds 3408–3411, are archived only in the prediction dataset.

C Additional Experimental Results

C.1 Full Reliability and Robustness Results

Table C1 lists per-seed event times, trajectory classes, final aligned and color-swap performance, and observation horizons for the main reliability sweep. The aggregation and censoring conventions are given in Appendix B.1.

Table C1: Per-seed shortcut dynamics across architectures and reliabilities. The main sweep contains all 60 trajectories.

Arch.	ρ	Seed	Class	T_f	$T_\times$	T_e	T_r	LI	Align	Swap	H
U-Net	0.80	3407	CF	cens.	cens.	500	500	N	0.9762	0.9686	5000
U-Net	0.80	3408	CF	cens.	cens.	400	400	N	0.9792	0.9779	5000
U-Net	0.80	3409	CF	cens.	cens.	400	350	N	0.9779	0.9781	5000
U-Net	0.80	3410	S→C	60	80	550	500	N	0.9758	0.9758	5000

Continued on next page

Table C1: Per-seed shortcut dynamics (continued).

Arch.	ρ	Seed	Class	T_f	$T_\times$	T_e	T_r	LI	Align	Swap	H
U-Net	0.80	3411	CF	cens.	cens.	450	450	N	0.9793	0.9788	5000
U-Net	0.95	3407	S→C	60	160	660	660	N	0.9762	0.9716	5000
U-Net	0.95	3408	S→C	70	110	550	550	N	0.9794	0.9531	5000
U-Net	0.95	3409	S→C	120	150	950	950	N	0.9747	0.9331	5000
U-Net	0.95	3410	S→C	100	170	690	690	N	0.9773	0.9383	5000
U-Net	0.95	3411	S→C	50	140	540	540	N	0.9782	0.9664	5000
U-Net	0.995	3407	S→C	60	2890	4470	4470	N	0.9776	0.9151	5000
U-Net	0.995	3408	S→C	170	1660	3750	3750	N	0.9777	0.6030	5000
U-Net	0.995	3409	S→C	70	1690	3630	3630	N	0.9780	0.9293	5000
U-Net	0.995	3410	S→C	500	2960	cens.	cens.	N	0.9785	0.5306	5000
U-Net	0.995	3411	S→C	50	1720	4730	4730	N	0.9787	0.7217	5000
U-Net	1.00	3407	LI	1700	cens.	cens.	cens.	Y	0.9777	0.0061	5000
U-Net	1.00	3408	LI	1550	cens.	cens.	cens.	Y	0.9795	0.0061	5000
U-Net	1.00	3409	LI	180	cens.	cens.	cens.	Y	0.9783	0.0059	5000
U-Net	1.00	3410	LI	190	cens.	cens.	cens.	Y	0.9787	0.0060	5000
U-Net	1.00	3411	LI	50	cens.	cens.	cens.	Y	0.9795	0.0060	5000
ResNet-18	0.80	3407	CF	cens.	cens.	250	250	N	0.9778	0.9775	5000
ResNet-18	0.80	3408	CF	cens.	cens.	310	310	N	0.9794	0.9777	5000
ResNet-18	0.80	3409	CF	cens.	cens.	220	220	N	0.9791	0.9780	5000
ResNet-18	0.80	3410	S→C	60	100	400	400	N	0.9768	0.9539	5000
ResNet-18	0.80	3411	CF	cens.	cens.	250	250	N	0.9789	0.9780	5000
ResNet-18	0.95	3407	S→C	100	170	730	730	N	0.9775	0.9583	5000
ResNet-18	0.95	3408	S→C	80	160	500	500	N	0.9785	0.9509	5000
ResNet-18	0.95	3409	S→C	80	160	420	420	N	0.9791	0.9734	5000
ResNet-18	0.95	3410	S→C	70	220	850	850	N	0.9796	0.8600	5000
ResNet-18	0.95	3411	S→C	100	130	530	530	N	0.9783	0.9559	5000
ResNet-18	0.995	3407	S→C	400	2130	4390	4390	N	0.9780	0.6542	5000
ResNet-18	0.995	3408	S→C	80	3440	cens.	cens.	N	0.9785	0.5307	5000
ResNet-18	0.995	3409	S→C	170	1340	cens.	cens.	N	0.9761	0.6227	5000
ResNet-18	0.995	3410	S→C	130	2950	cens.	cens.	N	0.9793	0.5583	5000
ResNet-18	0.995	3411	S→C	120	1870	cens.	cens.	N	0.9655	0.8708	5000
ResNet-18	1.00	3407	LI	400	cens.	cens.	cens.	Y	0.9786	0.0062	5000
ResNet-18	1.00	3408	LI	210	cens.	cens.	cens.	Y	0.9793	0.0061	5000
ResNet-18	1.00	3409	LI	170	cens.	cens.	cens.	Y	0.9792	0.0061	5000
ResNet-18	1.00	3410	LI	140	cens.	cens.	cens.	Y	0.9793	0.0060	5000
ResNet-18	1.00	3411	LI	1750	cens.	cens.	cens.	Y	0.9794	0.0059	5000
ViT-Tiny	0.80	3407	S→C	180	320	cens.	cens.	N	0.9448	0.8752	5000
ViT-Tiny	0.80	3408	S→C	100	350	cens.	cens.	N	0.9452	0.8815	5000
ViT-Tiny	0.80	3409	S→C	150	320	4800	4800	N	0.9460	0.9101	5000
ViT-Tiny	0.80	3410	S→C	90	410	4750	4750	N	0.9512	0.9123	5000
ViT-Tiny	0.80	3411	S→C	160	350	4200	4200	N	0.9513	0.8921	5000
ViT-Tiny	0.95	3407	S→C	180	1410	cens.	cens.	N	0.9549	0.8644	10000
ViT-Tiny	0.95	3408	S→C	170	1200	cens.	cens.	N	0.9499	0.7074	5000
ViT-Tiny	0.95	3409	S→C	350	840	11850	11850	N	0.9635	0.9398	20000
ViT-Tiny	0.95	3410	S→C	160	1050	9000	9000	N	0.9441	0.8668	10000
ViT-Tiny	0.95	3411	S→C	180	1430	cens.	cens.	N	0.9526	0.6716	5000
ViT-Tiny	0.995	3407	LI	190	cens.	cens.	cens.	Y	0.9434	0.1257	5000
ViT-Tiny	0.995	3408	LI	170	cens.	cens.	cens.	Y	0.9560	0.1404	5000
ViT-Tiny	0.995	3409	LI	80	cens.	cens.	cens.	Y	0.9539	0.0070	5000
ViT-Tiny	0.995	3410	LI	80	cens.	cens.	cens.	Y	0.9510	0.2561	5000
ViT-Tiny	0.995	3411	LI	210	cens.	cens.	cens.	Y	0.9560	0.0059	5000
ViT-Tiny	1.00	3407	LI	190	cens.	cens.	cens.	Y	0.9492	0.0061	5000
ViT-Tiny	1.00	3408	LI	90	cens.	cens.	cens.	Y	0.9518	0.0062	5000
ViT-Tiny	1.00	3409	LI	250	cens.	cens.	cens.	Y	0.9457	0.0060	5000
ViT-Tiny	1.00	3410	LI	90	cens.	cens.	cens.	Y	0.9524	0.0060	5000
ViT-Tiny	1.00	3411	LI	200	cens.	cens.	cens.	Y	0.9582	0.0059	5000

Notation. CF: core-first; S→C: shortcut-to-core; LI: shortcut lock-in; "cens." denotes a right-censored event, and H is the observation horizon. Median time to 90% aligned performance is 860

steps for ViT-Tiny versus 190 for ResNet-18; normalized by this time, the recovery-time gap shrinks to $5.7\times$.

C.2 Post-Focus Representation Dynamics

The matched ResNet-18 and ViT-Tiny analyses use the checkpoint alignment and measurements defined in Appendix B.2.

C.2.1 Cross-Architecture Evidence Transfer

Transferring masks that retain the top 25% of source evidence gives the following output fidelities:

	ResNet-18 target	ViT-Tiny target
ResNet-18 mask	0.870	0.849
ViT-Tiny mask	0.798	0.879

The transferred masks retain substantial task-relevant functionality in both architectures.

C.2.2 Temporal Representation Stability

The paired ResNet-minus-ViT temporal CKA difference is 0.0773, with 95% CI $[0.0628, 0.1008]$. This measure quantifies the greater temporal stationarity of ResNet-18 sample-level representations during post-focus recovery.

C.2.3 Sample-Geometry Recomposition

Held-out NRMSE is 0.421 for ViT-Tiny and 0.292 for ResNet-18; the paired ViT-minus-ResNet difference is 0.119, with 95% CI $[0.072, 0.160]$.

Sample-distance RSA is 0.846 for ViT-Tiny and 0.919 for ResNet-18, with every paired seed exhibiting greater sample-geometry stability for ResNet-18. Centroid-distance RSA is 0.981 for ViT-Tiny versus 0.927 for ResNet-18. Adjacent-checkpoint spatial-attention cosine similarity is 0.993 for ViT-Tiny and 0.981 for ResNet-18.

C.2.4 Selected Parameter-Group Rollback Analysis

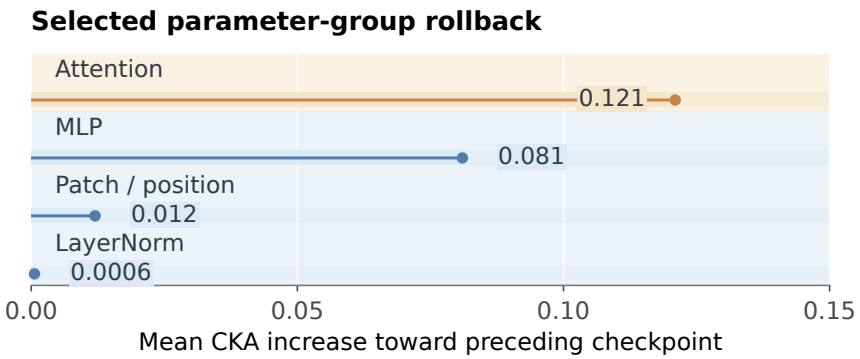


Figure 9: **Selected parameter-group rollback in ViT-Tiny.** Each selected group is restored to its value at the preceding frozen checkpoint while all other parameters remain fixed. Bars show the mean increase in CKA similarity to the preceding representation. Attention produces the largest increase among the tested groups.

Among the parameter groups tested, attention rollback produces the largest mean increase in similarity to the preceding representation (Figure 9). Mean recovery is 0.121 CKA for attention, 0.081 for the MLP, 0.012 for patch/position parameters, and 0.0006 for LayerNorm. Attention-minus-MLP recovery is 0.040, with 95% bootstrap CI $[0.024, 0.065]$, and attention exceeds MLP in all four seeds.

Parameter-drift normalization divides recovery by the corresponding parameter displacement, giving 4.47 for attention and 2.74 for the MLP.

The attention-associated representation change peaks near normalized post-focus recovery position $q = 0.8$ and is distributed across blocks 4, 5, 3, 7, and 6. LayerNorm changes are small and comparatively stabilizing. These measurements support attention-mediated sample-geometry recomposition as a distributed architecture-specific representation process.

C.2.5 Event-Aligned Corrective Transport

Table C2 summarizes gradient measurements aligned to T_{focus}, using the decision gap D of Equation 1 and the conflict transport $\mathcal{T}_c$ of Equation 2.

Table C2: **Event-aligned corrective transport at $\rho = 0.95$.** Decision gap and negative-cosine fraction summarize the five-seed $\rho = 0.95$ focus cohort. Exact signed conflict transport $\mathcal{T}_c$ is available only for the matched seed-3407 focus checkpoints of ResNet-18 and ViT-Tiny and is reported as a direct measurement rather than a population estimate.

Architecture	D at focus	Negative-cosine fraction	Exact $\mathcal{T}_c$
U-Net	-0.234	0.181	—
ResNet-18	-0.363	0.246	$+3.196$
ViT-Tiny	-0.617	0.397	-5.856

ViT-Tiny's corrective gradients contain substantially more mutually opposing directions (negative-cosine fraction 0.397 vs. 0.246). In the matched seed-3407 pair, full-network conflict transport at capture is $+3.196$ for ResNet-18 and -5.856 for ViT-Tiny: conflict updates initially move ViT-Tiny's counterfactual decision toward the shortcut. The matched seed illustrates the same architecture-dependent transport distinction seen in the five-seed gradient-direction statistics.

C.3 Training-Start Crossover Prediction

Four prospectively held-out ViT-Tiny runs at $\rho = 0.90$ (seeds 3408–3411) were not used to develop the original training-start predictor; their APEs are 8.17%, 34.15%, 6.41%, and 18.48%, for a median of 13.33%. We use this held-out median as the prospective timing result reported in the abstract. The 26-run retrospective cohort also contains these runs, evaluated with an optimized predictor whose configuration search included them (APEs 10.17%, 29.49%, 5.66%, and 16.82%).

For the 26 observed crossovers defined in Appendix B.6, median APE is 3.80% for U-Net, 9.46% for ResNet-18, and 5.64% for ViT-Tiny. The pooled median is 5.65% and the mean is 8.17%; 23/26 predictions (88.5%) lie within 15% of the observed crossover time.

Table C3: **Crossover-time prediction from the first ten optimizer updates.** The 26 runs use the consecutive-evaluation crossover criterion. APE is absolute prediction error divided by the observed T_{cross}. The last column counts runs with APE $\leq 15\%$. The pooled row weights each run equally.

Architecture	n	Median APE (%)	Mean APE (%)	Within 15%
U-Net	5	3.80	4.81	5/5
ResNet-18	6	9.46	10.94	5/6
ViT-Tiny	15	5.64	8.19	13/15
All	26	5.65	8.17	23/26

Table C4: **All 26 training-start crossover predictions.** Observed times use the three-consecutive-evaluation crossover rule of Appendix B.1. Predictions use steps 0–10; times are optimizer steps. APE uses unrounded predictions.

Architecture	ρ	Seed	Observed T_{cross}	Predicted	APE (%)
U-Net	0.80	3410	80	77.39	3.27
U-Net	0.90	3407	100	98.78	1.22
U-Net	0.90	3409	120	107.70	10.25
U-Net	0.90	3410	100	105.50	5.50
U-Net	0.90	3411	90	93.42	3.80
ResNet-18	0.80	3410	100	90.16	9.84
ResNet-18	0.90	3407	110	101.44	7.78
ResNet-18	0.90	3408	110	100.02	9.07
ResNet-18	0.90	3409	110	109.59	0.37
ResNet-18	0.90	3410	140	119.80	14.43
ResNet-18	0.90	3411	90	111.75	24.17
ViT-Tiny	0.70	3407	250	253.31	1.33
ViT-Tiny	0.70	3408	290	257.73	11.13
ViT-Tiny	0.70	3409	270	259.97	3.72
ViT-Tiny	0.70	3410	270	280.75	3.98
ViT-Tiny	0.70	3411	230	256.14	11.36
ViT-Tiny	0.80	3407	320	330.26	3.21
ViT-Tiny	0.80	3408	350	362.45	3.56
ViT-Tiny	0.80	3409	320	364.42	13.88
ViT-Tiny	0.80	3410	410	386.86	5.64
ViT-Tiny	0.80	3411	350	350.73	0.21
ViT-Tiny	0.90	3407	660	677.80	2.70
ViT-Tiny	0.90	3408	590	649.99	10.17
ViT-Tiny	0.90	3409	440	569.74	29.49
ViT-Tiny	0.90	3410	740	698.12	5.66
ViT-Tiny	0.90	3411	580	677.53	16.82

C.4 Complete Carrier Results

Table C5 reports the complete carrier comparison; Figure 10 shows the corresponding representation changes.

Table C5: **Controlled shortcut carriers on ShortcutSynthetic.** Exact event times are reported for the preregistered seed-3407 experiments at $\rho = 0.90$. Model, optimizer, equal-distance Hadamard codebook, and training protocol are fixed across carriers. Each carrier uses one seed.

Carrier	Support	T_{focus}	T_{cross}	Cross-to-escape	T_{escape}	Final P_{swap}	Final P_{removed}
Border	Local	75	3400	600	4000	0.953	0.957
Luminance	Global	50	1800	600	2400	0.892	0.877
Chromatic	Global	75	2000	400	2400	0.873	0.541

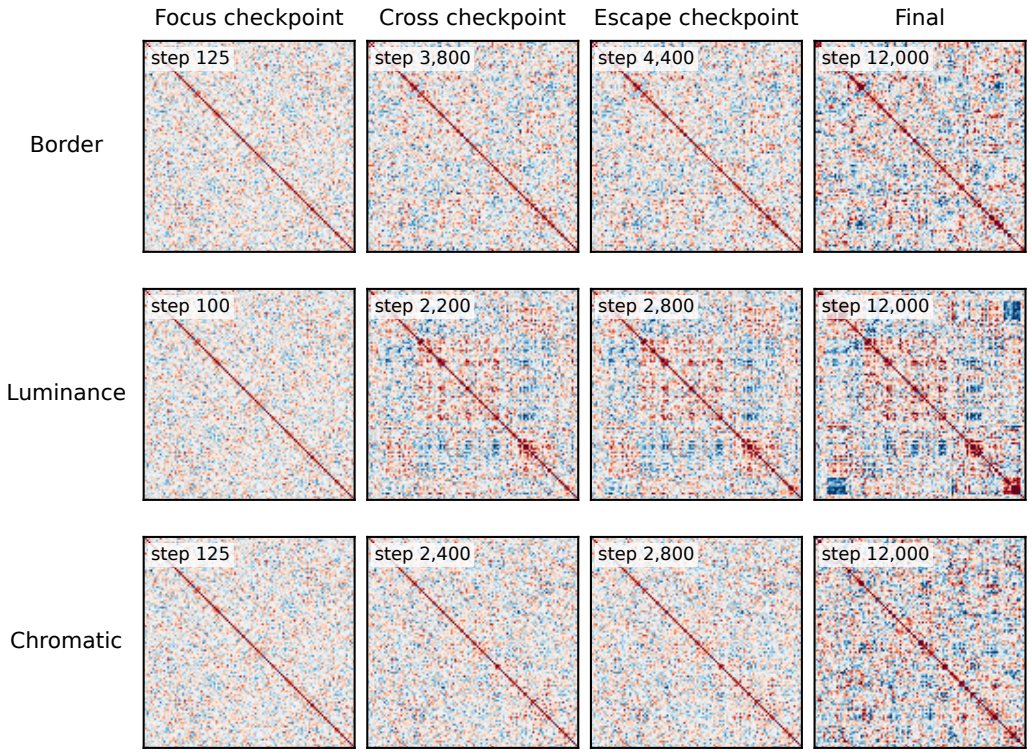


Figure 10: **Representation condensation across Hadamard carriers.** Pairwise cosine similarity in the final-block query projection is shown at the checkpoints labeled focus, crossover, escape, and final for border, luminance, and chromatic carriers. The panels compare representations at the displayed checkpoints, whose step labels differ from the event times in Table C5. The shared neuron order and color scale make the temporal reorganization directly comparable.

C.5 Complete Shallow Multimodal Cohort

The $\rho = 0.80$ escape risk set excludes one core-first run.

Table C6: **Controlled shallow multimodal shortcut dynamics.** F/C/E gives focus, crossover, and escape incidence; the escape denominator is the post-focus risk set. Event times are medians over observed events. Final text-swap accuracy is mean ± sample SD over five seeds. † denotes right-censoring without horizon imputation. The GQA fusion-depth experiment is reported separately in Table C7.

Model	Params	ρ	N	F/C/E	T_{focus}	T_{cross}	T_{escape}	Final text-swap
Shallow-MM	0.292M	0.80	5	4/5 / 4/5 / 4/4	15	80	210	0.903 ± 0.010
Shallow-MM	0.292M	0.95	5	5/5 / 5/5 / 5/5	20	390	1270	0.801 ± 0.031
Shallow-MM	0.292M	0.995	5	5/5 / 0/5 / 0/5	20	NR	NR†	0.332 ± 0.017
Shallow-MM	0.292M	1.00	5	5/5 / 0/5 / 0/5	20	NR	NR†	0.000 ± 0.000

C.6 GQA Depth Sweep and Recovery Trajectories

C.6.1 Full Depth Sweep and Trajectory Summaries

Early focus and variable recovery. Table C7 reports the complete five-seed sweep. Median focus occurs at steps 200–400 for every depth, whereas median escape ranges from 2,900 to 6,700

Table C7: **Five-seed GQA fusion-depth sweep at fixed $\rho = 0.75$.** Each fusion depth uses five independent seeds and a 20,000-step horizon. Behavioral crossover and escape events are recomputed from the same 100-step D_{beh} trajectories using the three-consecutive-evaluation rule. Event times are reported as median [IQR] when all runs are observed; full accuracy is mean ± sample SD. Parameters are approximate total model counts; times are optimizer steps. *One depth-1 run does not escape within the horizon and is right-censored at 20,000 steps; these cells report the Kaplan–Meier median and quartiles.

Depth	Params	T_{focus} med [IQR]	T_{cross} med [IQR]	T_{escape} med [IQR]	Cross-to-escape med [IQR]	Full acc. mean ± SD
1	24M	300 [200, 400]	1,300 [1,200, 1,600]	6,700 [6,700, 6,900]*	5,500 [5,300, 5,500]*	66.0 ± 0.2
2	25M	300 [200, 300]	1,100 [1,000, 1,200]	4,600 [4,400, 4,700]	3,300 [3,200, 3,800]	67.6 ± 0.4
4	26M	200 [200, 300]	1,000 [1,000, 1,200]	3,000 [3,000, 3,700]	2,100 [1,800, 2,400]	66.9 ± 0.5
8	29M	300 [200, 300]	1,300 [1,200, 1,300]	2,900 [2,800, 2,900]	1,600 [1,300, 1,600]	67.7 ± 0.3
12	32M	200 [200, 300]	1,300 [1,300, 1,300]	3,300 [2,700, 3,500]	2,000 [1,400, 2,100]	67.6 ± 0.4
14	34M	200 [200, 300]	900 [900, 1,300]	3,400 [3,200, 3,900]	2,300 [2,100, 3,000]	67.7 ± 0.3
16	36M	300 [200, 400]	1,100 [1,000, 1,200]	3,800 [3,500, 3,900]	2,600 [2,500, 2,600]	67.5 ± 0.4
18	37M	400 [300, 400]	1,300 [900, 1,300]	2,900 [2,700, 3,000]	1,700 [1,600, 1,800]	67.6 ± 0.3
20	39M	400 [400, 500]	1,200 [1,100, 1,300]	3,000 [3,000, 3,400]	2,200 [1,600, 2,500]	67.9 ± 0.5
22	40M	300 [200, 400]	1,300 [1,100, 1,500]	3,500 [3,500, 4,100]	2,600 [2,400, 2,900]	67.7 ± 0.7
24	42M	400 [400, 500]	1,400 [1,300, 1,500]	3,600 [3,500, 3,600]	2,300 [2,100, 2,500]	67.7 ± 0.3

steps. From depth 1 to depth 8, median escape advances from 6,700 to 2,900 steps, and the median crossover-to-escape interval contracts from 5,500 to 1,600 steps. Depth 1 is the only depth with a censored escape (4/5 runs observed), so its escape summaries are Kaplan–Meier estimates. Beyond the shallow regime, median escape varies nonmonotonically between 2,900 and 3,800 steps. The earliest median escape occurs at depths 8 and 18 (2,900 steps), whereas depth 14 has the earliest median crossover (900 steps) but an intermediate median escape (3,400 steps); crossover and sustained escape therefore order depths differently.

Event timing complements accuracy. Mean final accuracy spans 66.0–67.9%, with seed-level standard deviations of 0.2–0.7 percentage points, and the highest mean accuracy occurs at depth 20 (67.9%). Median escape time more than doubles across depths. Depths 2 and 8 reach similar mean accuracy (67.6% and 67.7%) but median escape at 4,600 and 2,900 steps, demonstrating the additional information provided by recovery time. In this experiment the operational escape event is the first of three consecutive evaluations with the behavioral gap $D_{\text{beh}}(t) = C(t) - A(t) \geq 0.20$. This threshold applies to decision shares, which differ from the continuous logit gap in Equation 1.

Trajectory-level recovery and later attenuation. Figure 11 is a seed-3407 trajectory visualization: 2,200 decision-gap values, 200 checkpoints per depth at 100-step intervals through step 20,000. Every sequence is shortcut-dominant around its focus annotation (steps 200–600) and remains core-dominant ($D_{\text{beh}} > 0$) at every sampled checkpoint from step 1,400 onward at all depths except depth 1, which does so from step 2,200. Across all five seeds, 46 of the 55 runs keep $D_{\text{beh}} \geq 0.20$ at step 20,000; 8 of the 54 runs that escape end below this margin. Their positive final gaps indicate attenuation within the core-dominant regime.

The heatmap distinguishes the onset of core preference, attainment of a stronger margin, and retention of that margin later in training. These stages reveal heterogeneous recovery paths.

Interpreting the depth trend. At fixed sampling reliability, depth and parameter count change together. Figure 7 shows a substantial shallow-to-intermediate-depth gain followed by fluctuations. Appendix A.4 relates a saturation law to explicit assumptions about the evidence budget and transport rate.

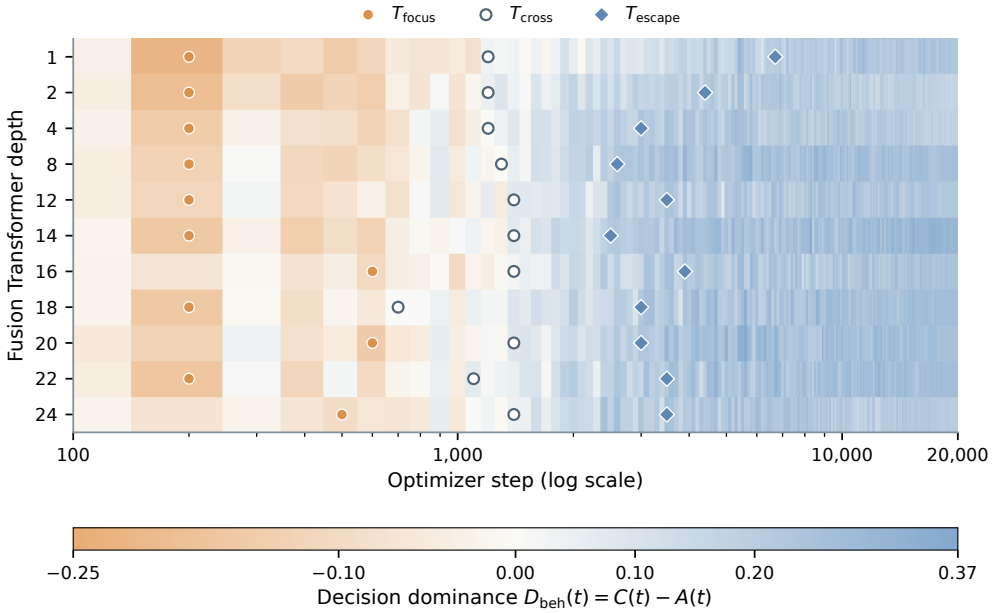


Figure 11: **The depth sweep reveals distinct shortcut-to-core trajectories (seed-3407 trajectory visualization).** Colors show the saved seed-3407 behavioral gaps at 100-step intervals on a logarithmic time axis; rows are the 11 fusion depths. Orange and blue indicate shortcut and core dominance. Symbols mark focus, crossover, and escape. Crossover and escape markers are computed directly from the displayed 100-step D_{beh} trajectory using the same three-consecutive-evaluation detector used throughout the GQA analysis. The color scale covers the full observed range.

C.6.2 Decision-Gap Heatmap and Margin Retention

Figure 11 shows the seed-3407 11×200 decision-gap matrix sampled at steps $100, 200, \ldots, 20{,}000$. Its 2,200 values span $[-0.219, 0.361]$, and the upper color limit of 0.37 accommodates the full range. The markers are the canonical events of these runs.

Margin retention after escape. In seed 3407, the final gap falls below 0.20 for depths 2 and 4 despite earlier escape events. Across the five-seed sweep, the number of runs that retain $D_{\mathrm{beh}} \geq 0.20$ at step 20,000 is 3, 4, 3, and 3 of 5 at depths 1, 2, 4, and 8, and 4–5 of 5 at every deeper depth. Thus positivity, a local three-checkpoint threshold event, and maintaining that threshold to the end of training are distinct observables. Later margin retention therefore provides information beyond the first escape event.

C.7 Numerical Verification

Independent numerical integration checks the local rates, invariants, and asymptotic coefficients (Figure 12). Maximum errors are 3.8×10^{-12} for the local core-product rate, 2.8×10^{-14} for the $\rho = 1$ invariant drift, and 2.97% for the tested asymptotic time coefficient.

Table C8: **Numerical verification of the exactly reducible model.** Closed-form rates, hitting-time coefficients, and the conserved quantity are compared against independent numerical integration.

Validated quantity	Error statistic	Result
Local core-product rate	maximum relative error	3.8×10^{-12}
Focus-time coefficient	maximum relative error	0.95%
Post-focus escape-time coefficient	maximum relative error	2.97%
Conserved quantity	maximum drift	2.8×10^{-14}

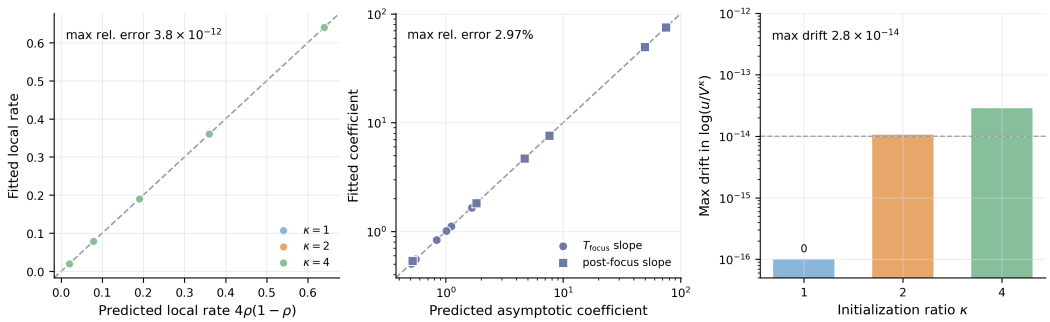


Figure 12: **Numerical verification of the exactly reducible model.** Predicted and fitted local rates, asymptotic time coefficients, and the $\rho = 1$ invariant agree to the errors stated in Appendix C.7. The identity line gives exact agreement.

D PROOFS

D.1 ARCHITECTURE-CONDITIONED TRANSPORT

Proposition 1 (Architecture-conditioned transport)**.** *Under gradient flow on* $L = \rho L_a + (1-\rho)L_c$,

$$\dot{D} = \rho \mathcal{T}_a + (1-\rho)\mathcal{T}_c, \qquad \mathcal{T}_j = -g_D^\top K_{\mathcal{A}} r_j, \qquad K_{\mathcal{A}} = J_{\mathcal{A}} J_{\mathcal{A}}^\top. \tag{30}$$

If $D(t_e) = b$*, integration from* t_f *to* t_e *gives Eq. 4.*

Proof. Apply the chain rule and substitute $\dot{\theta} = -\nabla L$, $\nabla_\theta D = J_{\mathcal{A}}^\top g_D$, and $\nabla_\theta L_j = J_{\mathcal{A}}^\top r_j$; integration gives the budget. □

Corollary 1 (Transport exponent)**.** *Let* $q = 1 - \rho$*. If* $0 < c_1 \le B_{\text{esc}} \le c_2 < \infty$ *and* $d_1 q^{1-\alpha} \le \bar{v} \le d_2 q^{1-\alpha}$ *as* $q \downarrow 0$*, then* $T_{\text{escape}} - T_{\text{focus}} = \Theta(q^{-(1-\alpha)})$.

Proof. Apply Eq. 5 and bound its numerator and denominator by the assumed constants. □

D.2 FROM TRAJECTORY APPROXIMATION TO CROSSING-TIME ERROR

Let $z(t)$ be a measured network state, $\dot{z} = F(z) + e(t)$, and let z_r solve $\dot{z}_r = F(z_r)$. If F is L-Lipschitz, $\|e(t)\| \le \epsilon$, and the initial mismatch is δ_0, then

$$\Delta_z(T) = \sup_{t_f \le t \le T} \|z(t) - z_r(t)\| \le \delta_0 e^{L\tau} + \frac{\epsilon}{L}(e^{L\tau} - 1), \quad \tau = T - t_f. \tag{31}$$

For an L_D-Lipschitz decision gap crossing transversely at speed at least m, the induced prediction obeys

$$|\tau_b - \tau_b^{(r)}| \le (L_D/m)\Delta_z(T). \tag{32}$$

Under these conditions, the bound converts reduced-state approximation error into continuous crossing-time error. For the empirical forecasts in Section 4.2, applying the bound entails checking these conditions and accounting for checkpoint spacing and consecutive confirmation.

Theorem 1 (Network-to-reduced escape-time transfer)**.** *If* $\dot{z} = F(z) + e$, $\dot{z}_r = F(z_r)$, F *is* L*-Lipschitz,* $\|e\| \le \epsilon$*, and the initial error is* δ_0*, then Eq. 31 holds. For an* L_D*-Lipschitz gap with a unique crossing of slope at least* $m > 0$*, Eq. 32 holds.*

Proof. Gronwall applied to $z - z_r$ gives the trajectory bound. The gap mismatch is at most $L_D \Delta_z(T)$; the mean-value theorem divides it by the crossing slope m to give the time bound. □

D.3 Balanced Product Routes

Let $U = \kappa \prod_{r=1}^{L_s} u_r$ and $V = \prod_{r=1}^{L_c} v_r$. Under balanced positive initialization, symmetry of the coordinate ODEs preserves equality within each route. Consequently

$$\dot{U} = \sum_r \frac{U}{u_r} \dot{u}_r = L_s \kappa^{2/L_s} U^{2-2/L_s} (-\partial_U L), \tag{33}$$

and the same calculation gives Eq. 8 for V.

Proposition 2 (Finite-horizon lock-in for $\rho < 1$)**.** *For $1/2 < \rho < 1$ and $U(0), V(0) > 0$, solutions of Eq. 8 obey $U(t) \le \max\{U(0), U^\star\}$ and $V(t) \to \infty$, where $U^\star = \log(\rho/(1-\rho))$. Thus $V(t) - U(t)$ crosses every finite threshold.*

Proof. Γ_V is strictly positive, so V is increasing. Put $q = 1 - \rho$ and $r = e^{U^\star} = \rho/q$. On $U = U^\star$ and $V \ge 0$,

$$\frac{\rho}{1 + re^V} \le \frac{q}{1 + e^V/r}, \tag{34}$$

because this inequality reduces to $(\rho - q)(1 - e^V) \le 0$. Hence the vector field cannot cross outward through $U = U^\star$, which proves the bound on U. If V had a finite limit, then (U, V) would remain in a compact set with $V \ge V(0) > 0$. On that set the positive continuous factor multiplying Γ_V has a positive lower bound, contradicting convergence of V. Therefore $V \to \infty$; boundedness of U then proves the claim. □

Proposition 3 ($\rho = 1$ invariant for depth two)**.** *For balanced depth-two factors u and v at $\rho = 1$, with shortcut route scale κ, u/v^κ is constant along the flow. At $\kappa = 1$ the sign of the hard-decision gap is preserved.*

Proof. Both routes receive the same scalar logistic residual. The factor equations therefore give $\dot{u}/u = \kappa \dot{v}/v$. Integration yields $\log u - \kappa \log v = \text{constant}$. At $\kappa = 1$, this invariant preserves the ordering of the two routes and hence the hard decision. □

D.4 Small-Initialization Asymptotics

Theorem 2 (Focus and escape times)**.** *Let $L_s = L_c = 2$, $U(0) = \kappa c_s \varepsilon^2$, $V(0) = c_c \varepsilon^2$, and $a = \kappa(2\rho - 1) > 1$. If the operational focus threshold lies below $U^\star$, Eq. 12 holds as $\varepsilon \downarrow 0$; changing between fixed crossover and escape thresholds changes only the $O(1)$ term.*

Proof. In the small-margin region the logistic residuals are $1/2 + O(U + V)$. Hence $\dot{U} = aU + O(U^2 + UV)$ and $T_{\text{focus}} = (2/a)\log(1/\varepsilon) + O(1)$. During this interval the core route grows at its slower linear rate, so on entry to the post-focus region $V_{\text{in}} = \Theta(\varepsilon^{2(1-1/a)})$. Once U reaches its finite balance scale, the leading core equation is $\dot{V} = 4\rho(1-\rho)V + O(V^2)$ until V is order one. Therefore

$$\int_{V_{\text{in}}}^{O(1)} \frac{dV}{4\rho(1-\rho)V + O(V^2)} = \frac{1 - 1/a}{2\rho(1-\rho)} \log(1/\varepsilon) + O(1), \tag{35}$$

which proves the post-focus expression. □