Learning Discrete Successor Transitions in Continuous Attractor Networks: Emergence, Limits, and Topological Constraints

Daniel Brownell
Independent Researcher
Cape Town, South Africa
daniel.brownell@gmail.com

Abstract

Continuous attractor networks (CANs) are a well-established class of models for representing low-dimensional continuous variables such as head direction, spatial position, and phase [1, 2, 3, 4]. In canonical spatial domains, transitions along the attractor manifold are driven by continuous displacement signals—such as angular velocity—provided by sensorimotor systems external to the CAN itself. When such signals are not explicitly provided as dedicated displacement inputs, it remains unclear whether attractor-based circuits can reliably acquire recurrent dynamics that support stable state transitions, or whether alternative predictive strategies dominate.

In this work, we present an experimental framework for training CANs to perform successor-like transitions between stable attractor states in the absence of externally provided displacement signals. We compare two recurrent topologies—a circular ring and a folded "snake" manifold—and systematically vary the temporal regime under which stability is evaluated. We find that, under short evaluation windows, networks consistently converge to impulse-driven associative solutions that achieve high apparent accuracy yet lack persistent attractor dynamics. Only when stability is explicitly enforced over extended free-run periods do genuine attractor-based transition dynamics emerge. This suggests that shortcut solutions are the default outcome of local learning in recurrent networks, while attractor dynamics represent a constrained regime rather than a generic result.

Furthermore, we demonstrate that topology strictly limits the capacity for learned transitions. While the continuous ring topology achieves perfect stability over long horizons ($t = 120$), the folded snake topology hits a geometric limit characterized by failure at manifold discontinuities, which neither curriculum learning nor basal ganglia–inspired gating can fully overcome.

***K*eywords** continuous attractor networks · successor representation · basal ganglia · curriculum learning · shortcut learning

1 Introduction

Continuous attractor networks (CANs) provide a principled mechanism for representing continuous variables through localized "bump" activity stabilized by recurrent excitation and inhibition [1]. Such networks have been studied extensively as models of head direction cells [2], grid cells [3], and eye position integrators [5]. A defining property of CANs is the existence of a low-dimensional manifold of approximately equivalent stable states.

In canonical spatial domains, transitions along this manifold are driven by continuous displacement signals provided by external sensorimotor systems [6, 7]. Under these conditions, the CAN functions primarily as a state integrator. However, many cognitive operations require consistent transitions along an internal representational manifold, even when no explicit displacement signal is provided to drive those transitions.

This distinction raises a fundamental question: to what extent can local, attractor-based circuits learn not only stable state representations but also reliable state transitions in the absence of externally provided displacement signals?

Related proposals argue that grid-cell-like representations and displacement-style operations may generalize beyond spatial domains, potentially providing a cortical substrate for internal simulation [8, 9]. While successor representations have been extensively studied in reinforcement learning [10, 11, 12], these approaches typically abstract away the underlying neural dynamics. Conversely, much of the CAN literature assumes that transition dynamics are hard-coded or strictly sensory-driven [13].

Crucially, we argue that predictive success does not imply attractor dynamics. We investigate the conditions under which transient predictive activity can be distinguished from genuine stable displacements, identifying a pervasive tendency for networks to learn “shortcut” solutions rather than robust attractors [14].

For clarity, learning in all experiments refers to experience-driven modification of recurrent connectivity under externally defined training signals, rather than autonomous planning or action selection.

2 Model Overview

2.1 Continuous Attractor Dynamics

Each model consists of a recurrent neural population whose activity evolves according to leaky integrator dynamics with lateral excitation and inhibition, consistent with classical neural field formulations [1, 13]. Stable attractor states correspond to localized bumps of activity on a low-dimensional manifold. We consider two manifolds:

- **Ring topology**: A circular manifold with wraparound connectivity, providing a continuous path for displacement [2].
- **Snake topology**: A folded linear manifold embedded in two dimensions, introducing boundary effects and anisotropic transition geometry.

2.2 Successor Transitions as Learned Displacements

Successor transitions are implemented as learned recurrent projections that bias the network toward the next attractor state. Functionally, these projections act as learned displacement operators on the attractor manifold, analogous to asymmetric connectivity induced by velocity signals in spatial CANs [3].

2.3 Gating and Control

We evaluate two control policies that regulate when state transitions may occur. In the *Standard* controller, transitions are driven directly by recurrent dynamics without any explicit gating signal. In the *BG-gated* controller, an additional suppressive mechanism—loosely inspired by the basal ganglia—can delay or inhibit transitions, allowing the network additional time to settle before displacement dynamics are engaged [15, 16, 17].

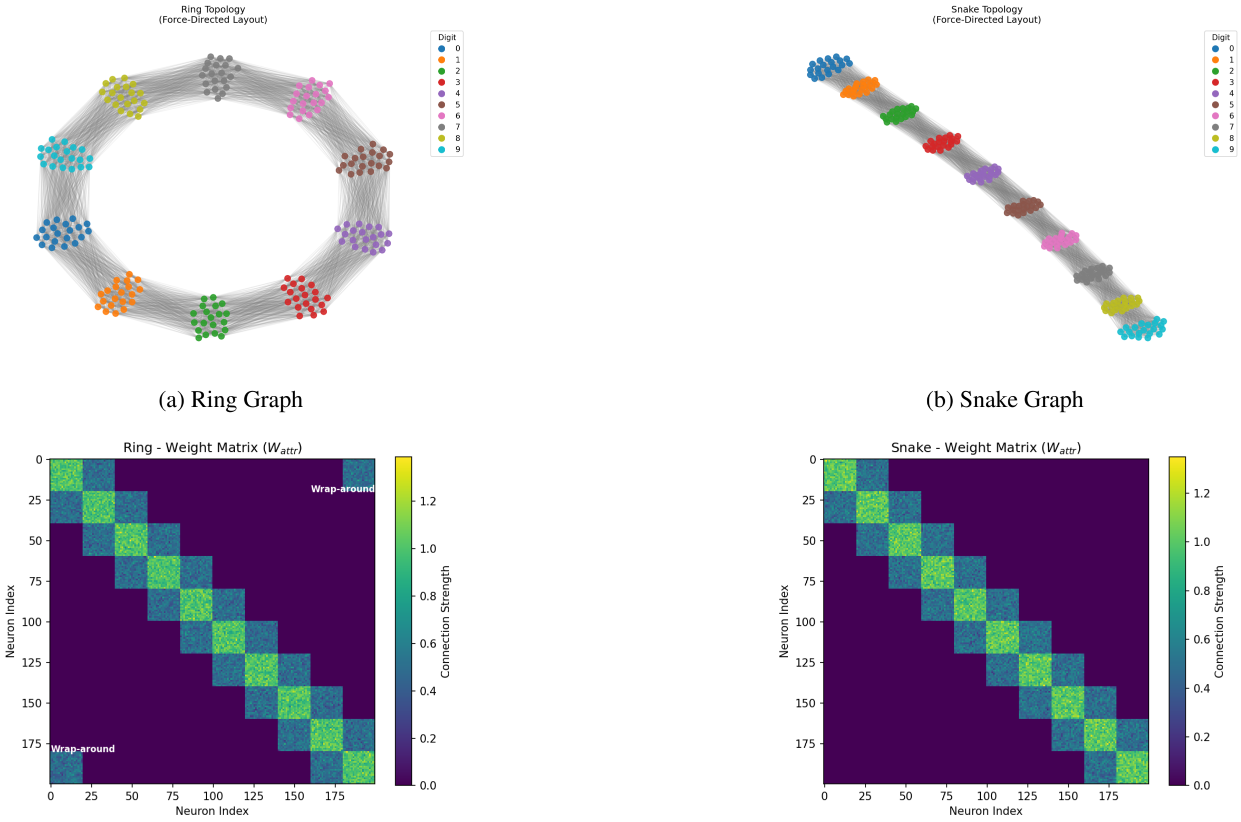
Importantly, the BG controller does not introduce new representational structure or alter the underlying attractor dynamics. Rather, it provides an orthogonal control signal that regulates the timing of transitions. Both controllers are evaluated under multiple transition-evaluation regimes described below.

2.4 Training Curriculum and Evaluation Tasks (Phases A–C)

Curriculum structure. Training is organized as a three-phase curriculum in which representational stability is learned before successor transition dynamics are optimized. All experiments share the same underlying recurrent CAN architecture and differ only in topology (Ring vs. Snake), controller (Standard vs. BG-gated), and the temporal regime used for scoring and selection.

Phase A (Digit attractor formation). In Phase A the network is trained to represent ten discrete symbolic states (digits 0–9) as stable bump attractors on the manifold. Training adjusts recurrent dynamics so that each digit cue reliably converges to a corresponding stable attractor state. Phase A is considered complete when per-digit readout accuracy reaches 100% on a validation procedure; we then export a Phase A snapshot (learned recurrent state and prototypes) which is used to initialize downstream training.

Phase B (Counting curriculum; optional). Phase B is an intermediate curriculum stage intended to shape sequential dynamics on the manifold before training direct successor transitions. In the *countingOn* condition, Phase B trains the network under a structured traversal objective consistent with counting. In the ablation *countingOff*, Phase B is skipped and Phase C begins immediately from the Phase A snapshot. This comparison tests whether an explicit



(a) Ring Graph

(b) Snake Graph

(c) Ring Weights

(d) Snake Weights

Figure 1: **Topology Comparisons.** Force-directed graphs (top) and weight matrices (bottom) revealing the structural differences between Ring and Snake manifolds. Note the continuous connectivity in the Ring matrix versus the hard boundary cutoffs in the Snake matrix.

intermediate traversal curriculum changes the optimization path for successor learning, particularly in topologies where local geometry may bias the network toward shortcut solutions.

Phase C (Successor transitions). Phase C trains the network to implement successor-like transitions (e.g., $x \mapsto x + 1$ modulo 10) as learned displacements on the attractor manifold. After an initial digit state is established, the model is required to transition to the successor state and maintain that state autonomously during a free-run period. Phase C admits two solution classes: short-horizon impulse-like transients versus long-horizon self-sustaining attractors.

Controllers. Both Standard and BG-gated controllers operate on the same underlying recurrent network. The Standard controller allows transitions to proceed immediately under learned recurrent dynamics. The BG-gated controller adds a suppressive timing signal that can delay or inhibit transitions (e.g., waiting steps and optional action sampling), intended to provide additional settling time before displacement dynamics are engaged. In all cases, the controller modulates *when* transitions are permitted rather than altering the Phase A representational space.

2.5 Automated Hyperparameter Search (Megatune Phase B/C)

Motivation. The Phase C task admits both (i) shortcut solutions that briefly activate the target state but decay during free-run and (ii) attractor-consistent solutions that sustain the target state over extended horizons. When evaluation is restricted to short temporal windows near transition onset, both solution classes can achieve similarly high prediction accuracy. As a result, naïve training and model selection procedures systematically favor transient solutions that satisfy the loss function without establishing stable attractor dynamics. To make comparisons between topologies and controllers meaningful, we therefore require an automated procedure that explicitly rejects transient solutions during hyperparameter search.

Megatuner overview. We use an automated multi-stage search procedure (`megatune-phaseBC`) that tunes parameters governing attractor dynamics (e.g., inhibition strength k_{inhib}, threshold v_{th}, leak/decay α, and row-normalized weight magnitudes) and transition-learning parameters (e.g., transition learning rate η_{trans}), along with the activity scoring mode used to measure stability (e.g., EMA-based spike activity) and its smoothing parameter. The megatuner is run per topology (Ring and Snake) and per controller family where applicable, producing best-configuration artifacts (e.g., `megatune_best_ring.json`, `megatune_best_snake.json`) that are then used for verification runs and diagnostic plots.

Temporal enforcement and windowing. Each transition trial is divided into a transition period of length t_{trans} followed by a free-run tail of length t_{tail}. Candidate configurations are scored over configurable temporal windows (e.g., mean over the whole trial versus late windows that isolate post-transient dynamics). To suppress impulse-based "cheating," attractor-enforced settings (a) exclude the first k time steps after transition onset from evaluation and (b) measure similarity to the target prototype only during the late phase of the trial, after external drive has subsided. This forces candidate solutions to maintain the state representation via recurrent dynamics rather than a brief associative pulse.

Composite objective and constraints. Candidates are scored with a composite objective trading off (1) successor accuracy and, where applicable, (2) sustained stability during free-run (*sustain*), together with penalties for (3) pathological failure modes such as collapse, silence, or aborted trials. The sustain term is only applied in long-horizon, attractor-enforced regimes (e.g., $t_{\text{trans}} = 120$), and is omitted in short-horizon settings where persistence cannot be meaningfully assessed. In addition to weighted scoring, we enforce hard constraints when required, including disallowing impulse-based categorization and requiring that learning and evaluation windows match (preventing optimization on an easier training window than the one used for reporting).

Staged search and verification. The megatuner uses staged filtering to allocate compute efficiently: an initial guardrail stage rejects obviously unstable candidates with a small number of trials, and later stages evaluate more trials while retaining only a top-K subset. The final selected configuration is then validated by (i) an independent probe step that evaluates candidates under resolved window settings and (ii) full Phase B/C training runs under *countingOn* and *countingOff* conditions, which generate the accuracy histories, confusion matrices, and transition trace plots reported in Results.

Relationship to reported regimes. We intentionally contrast megatuner settings corresponding to distinct dynamical regimes. Short-horizon settings (e.g., $t_{\text{trans}} = 40$, $t_{\text{tail}} = 10$, mean-window scoring, impulse allowed) often yield high apparent accuracy while permitting transient solutions. Long-horizon attractor-enforced settings (e.g., $t_{\text{trans}} = 120$, $t_{\text{tail}} = 30$, late-window scoring, exclude-first-k steps, impulse disallowed) bias the search toward genuine attractor dynamics and underpin the Ring vs. Snake comparison presented here.

3 Results

Unless stated otherwise, Phase B/C parameters were selected by the megatuner under the specified temporal regime (short-horizon vs. attractor-enforced long-horizon) and then verified by independent probe and full training runs.

3.1 Associative Shortcut Dynamics as the Default Regime

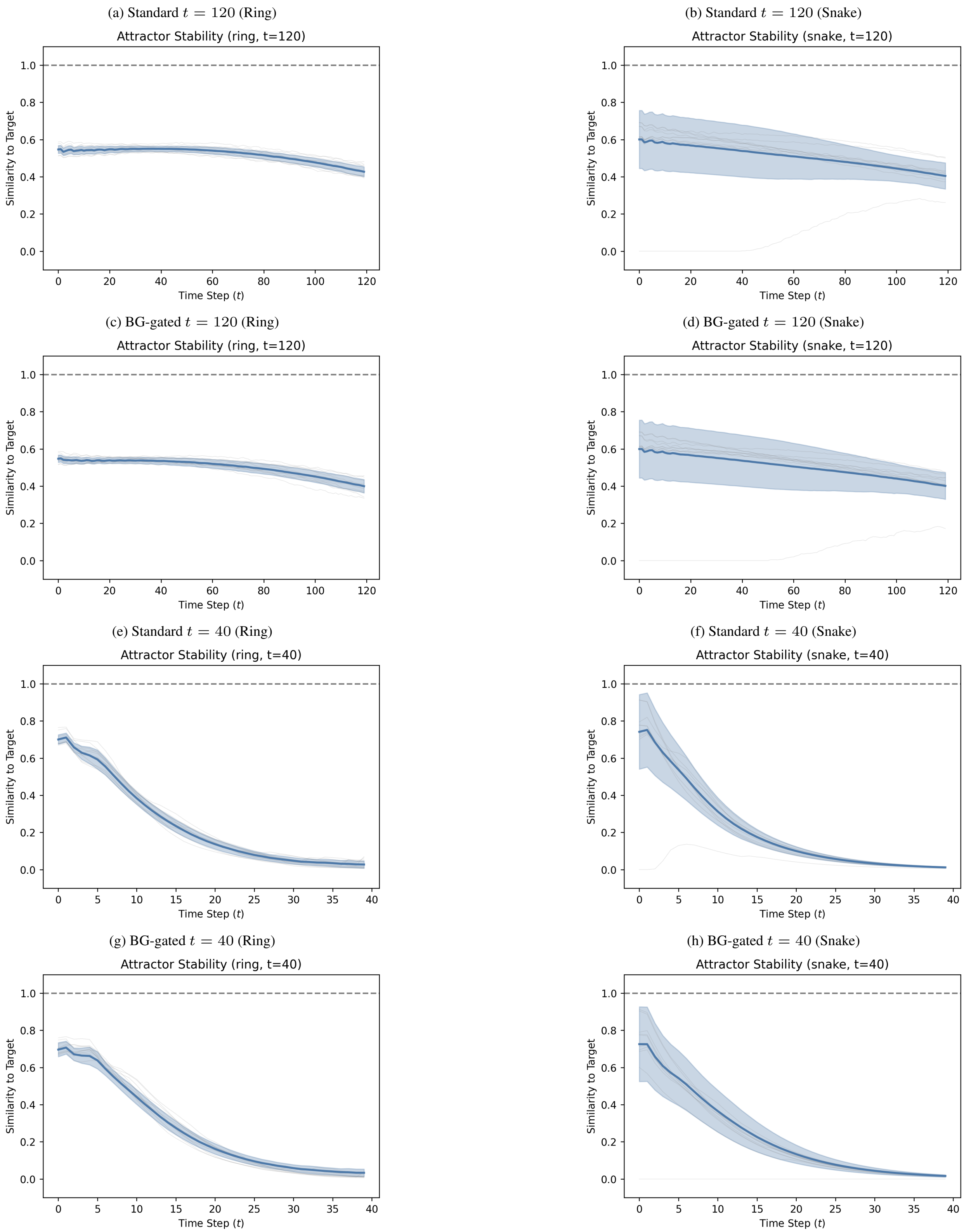
We observe that under standard training conditions with short evaluation windows, models consistently achieve high readout accuracy without establishing stable attractors. In this regime, the network learns an impulse-driven association: the initial state triggers a transient spike of activity at the target location, which satisfies the prediction loss but decays rapidly once the input drive is removed.

We term this "Impulse Cheating" or Associative Shortcut Dynamics. This is not a rare failure mode but the default outcome of gradient descent on predictive tasks in recurrent networks. The network minimizes energy by predicting the output without investing in the recurrent weights necessary to sustain the activity bump against noise.

3.2 Regime Selection via Composite Constraints

Given this tendency, stable attractors require evaluation that rejects transients; increasing the horizon alone (e.g., $t = 40 \to 120$) is insufficient without late-window scoring.

Therefore, we define an **Attractor Regime** that disallows impulse-based categorization, excludes the first $k = 8$ steps after transition onset, and scores accuracy only in the late phase after external drive has subsided. This forces candidate solutions to maintain the state via recurrent dynamics rather than a brief associative pulse, aligning with curriculum learning principles [18].



(a) Standard $t = 120$ (Ring)

(b) Standard $t = 120$ (Snake)

(c) BG-gated $t = 120$ (Ring)

(d) BG-gated $t = 120$ (Snake)

(e) Standard $t = 40$ (Ring)

(f) Standard $t = 40$ (Snake)

(g) BG-gated $t = 40$ (Ring)

(h) BG-gated $t = 40$ (Snake)

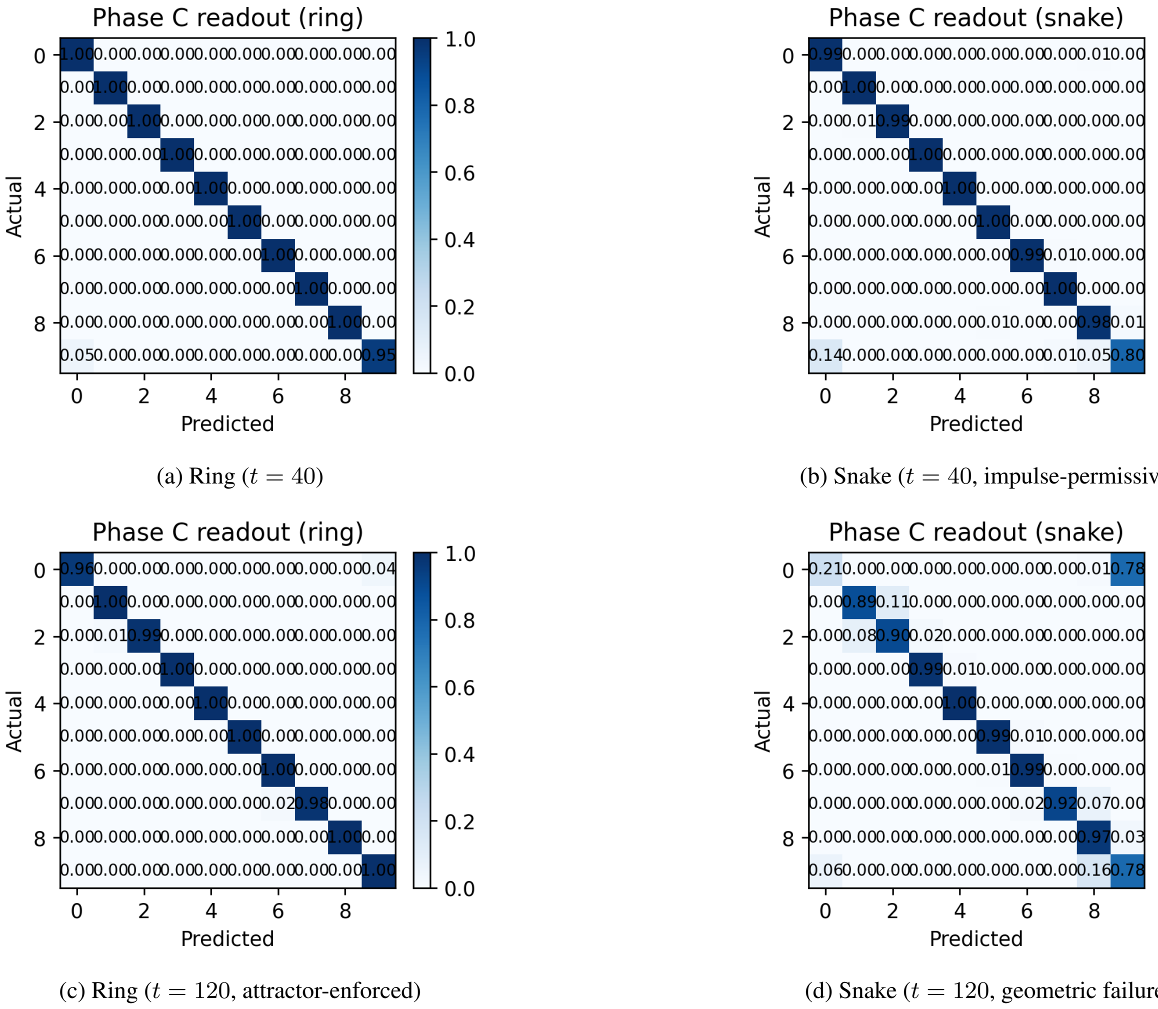
Figure 2: **Transition stability traces across horizons and controllers.** Cosine similarity to the target prototype during free-run dynamics after a successor transition. Short-horizon evaluation ($t = 40$) permits impulse-dominated transients, while long-horizon evaluation ($t = 120$) reveals robust Ring stability and persistent Snake drift that BG gating does not eliminate.

3.3 The Geometric Limit

The failure of the Snake topology reflects a structural constraint rather than an optimization or control failure. Crucially, this limit only becomes visible once transient shortcut solutions are excluded. Under short-horizon evaluation regimes ($t_{\text{trans}} = 40$), the Snake topology achieves near-perfect Phase C accuracy by exploiting impulse-driven dynamics that briefly activate the correct successor state without sustaining it. In contrast, under attractor-enforced long-horizon regimes ($t_{\text{trans}} = 120$), these shortcut solutions are no longer sufficient, revealing a systematic geometric failure.

Figure 3 compares Phase C confusion matrices for the Standard controller under short-horizon ($t = 40$) and long-horizon ($t = 120$) regimes. At $t = 40$, both Ring and Snake appear successful: the Snake confusion matrix is nearly diagonal, despite relying on transient activity that decays during free-run. At $t = 120$, however, the Snake exhibits persistent off-diagonal errors concentrated at manifold discontinuities (most prominently the wrap-around transition $9 \to 0$), while the Ring remains perfectly diagonal.

This divergence demonstrates that the Snake topology supports correct successor *classification* under impulse-cheating regimes but cannot sustain correct successor *dynamics* once stability is enforced. The failure is therefore geometric: the folded manifold introduces non-local discontinuities that cannot be bridged by locally learned recurrent displacements, regardless of curriculum, gating, or extended settling time.



(a) Ring ($t = 40$)

(b) Snake ($t = 40$, impulse-permissive)

(c) Ring ($t = 120$, attractor-enforced)

(d) Snake ($t = 120$, geometric failure)

Figure 3: **Impulse success versus geometric failure in Phase C.** Under short-horizon evaluation ($t = 40$), both Ring and Snake achieve high apparent accuracy, despite the Snake relying on transient impulse dynamics. Under long-horizon attractor-enforced evaluation ($t = 120$), the Ring remains perfectly stable, while the Snake exhibits systematic errors at topological discontinuities (notably $9 \to 0$), revealing a structural geometric limit.

3.4 Control vs. Geometry

We compared the Standard controller against a BG-gated controller across both horizons and curricula (Table 1). In the attractor-enforced long-horizon regime ($t = 120$), the Snake topology saturates at the same ceiling under both controllers (0.88 with countingOn), indicating a geometric bottleneck rather than insufficient settling time. Moreover, BG gating does not reliably improve performance and can degrade it in some settings (notably countingOff), reinforcing that timing control alone cannot compensate for topological discontinuities.

Topology	Controller	Curriculum	Horizon	Acc.	Regime
Ring	Standard	countingOn	$t = 40$	1.000	Short-horizon
Ring	Standard	countingOff	$t = 40$	1.000	Short-horizon
Snake	Standard	countingOn	$t = 40$	1.000	Short-horizon
Snake	Standard	countingOff	$t = 40$	0.945	Short-horizon
Ring	Standard	countingOn	$t = 120$	1.000	Robust attractor
Ring	Standard	countingOff	$t = 120$	1.000	Robust attractor
Snake	Standard	countingOn	$t = 120$	0.880	Geom. limit
Snake	Standard	countingOff	$t = 120$	0.840	Geom. limit
Ring	BG-gated	countingOn	$t = 40$	1.000	Short-horizon
Ring	BG-gated	countingOff	$t = 40$	1.000	Short-horizon
Snake	BG-gated	countingOn	$t = 40$	0.960	Short-horizon
Snake	BG-gated	countingOff	$t = 40$	0.280	Control fail
Ring	BG-gated	countingOn	$t = 120$	1.000	Robust attractor
Ring	BG-gated	countingOff	$t = 120$	0.365	Degraded
Snake	BG-gated	countingOn	$t = 120$	0.880	Geom. limit
Snake	BG-gated	countingOff	$t = 120$	0.720	Geom. limit

Table 1: **Control vs. geometry across controllers, curricula, and horizons (Phase C).** At $t = 40$, accuracy reflects short-window success and does not certify persistent attractor dynamics (*Short-horizon*). At $t = 120$ under attractor-enforced evaluation, Ring exhibits robust persistence, while Snake is limited by manifold discontinuities (*Geom. limit*). BG gating does not rescue Snake and can degrade performance in some settings (*Control fail*, *Degraded*).

4 Discussion

This study began with a simple question: to what extent can a local recurrent circuit learn not only stable state representations, but also reliable state transitions when no explicit displacement signal is provided. In spatial CAN models, transitions are typically driven by an external velocity or displacement input; here we asked whether comparable transition dynamics can be acquired from training alone, and how easily one can confuse "correct outputs" with "correct dynamics" when evaluating such systems.

Prediction accuracy can hide non-attractor solutions. Across conditions, we found that Phase C admits at least two qualitatively different solution classes. Under short-horizon, impulse-permissive regimes, the network can achieve high readout accuracy by producing a brief associative pulse at the target state. This looks like a successful successor transition when measured near transition onset, but it does not imply that the target state is stably maintained during free-run. The key methodological point is therefore not merely that "longer trials" matter, but that *selection criteria must explicitly reject early transients* (e.g., by excluding the first k steps and scoring only late-trial similarity). Without those constraints, naïve training and selection procedures systematically overestimate success by rewarding solutions that satisfy the loss function transiently rather than sustaining the representational state.

Implications for architectural extensions. A natural response to the geometric failure observed in the Snake topology is to posit additional non-local mechanisms capable of bridging discontinuities in the representational manifold. In biological models, this role is often attributed to active dendritic processes or other forms of context-dependent, long-range coupling. A concrete instantiation of this idea in the present setting would be to relax strict locality by introducing sparse long-range connectivity (e.g., a small fraction of background edges) that could couple otherwise disconnected regions of the manifold. Such connectivity might partially mitigate fold-induced failures by providing non-local influence paths, but it would also alter the effective attractor geometry and risks reintroducing shortcut dynamics unless carefully constrained.

From this perspective, our results do not argue against the necessity of additional mechanisms; rather, they help clarify what those mechanisms would need to accomplish. The failure modes identified here arise when successor transitions are delegated entirely to local recurrent attractor dynamics. Any architectural extension that successfully rescues these transitions must therefore implement conditional or non-local displacement structure beyond what a simple CAN provides. In this sense, the present results delineate a boundary: they show which transition behaviors cannot be obtained from local attractor recurrence alone, and thereby sharpen the functional role that additional architectural features would have to play.

Attractor-consistent transitions emerge, but only in a constrained regime. When evaluated under attractor-enforced settings (late-window scoring, impulse disallowed), the Ring topology supports robust successor transitions and persistent bump stability across long horizons. In that regime, the same network and learning rules that happily converge to shortcut solutions can also support genuine attractor-consistent dynamics, suggesting that the difficulty is not an inherent incapacity to form stable transitions, but the ease with which optimization discovers cheaper predictive strategies unless the task is formulated to rule them out.

Topology imposes hard limits on learned transitions. The Ring versus Snake comparison shows that even when shortcut solutions are excluded, topology strongly constrains what transitions can be learned by local recurrence. The Snake manifold, which preserves local adjacency but introduces discontinuities at fold boundaries, exhibits systematic errors concentrated at those discontinuities once attractor stability is enforced (notably the $9 \to 0$ wrap-around transition). This pattern is consistent with a geometric bottleneck: locally learned displacement-like dynamics can slide activity along continuous neighborhoods, but cannot reliably implement non-local "jumps" across a fold without additional structure or inputs. In this sense, the Snake results help separate two failure modes that can otherwise be conflated: transient cheating (an evaluation artifact) versus genuine geometric limitation (a representational constraint).

Control mechanisms modulate timing, not geometry. We tested a BG-gated controller intended to provide additional settling time before transitions are engaged. In the attractor-enforced regime, BG gating did not raise the Snake ceiling and in several settings degraded performance, suggesting that timing control alone cannot compensate for discontinuities that require non-local transitions. This does not rule out more complex control schemes, but it supports a narrower conclusion: within the class of controllers tested here, the dominant limitation for the Snake topology is geometric rather than temporal.

Implications and scope. Our results are deliberately modest: they do not claim that cortical circuits cannot realize CAN-like dynamics, nor do they prescribe a particular biological mechanism. Instead, they identify a practical and conceptual hazard: recurrent networks trained for successor-like prediction will often adopt transient predictive strategies that are difficult to distinguish from attractor dynamics unless evaluation is designed to measure persistence. This matters for any theory that leans on CAN-like modules for internally generated trajectories (e.g., imagined transitions, replay, or "thought experiments") in the absence of continuous sensory displacement input. If such circuits are to serve as stable internal simulators, then either (i) additional mechanisms must bias learning toward sustained attractors, or (ii) the relevant computations may rely on dynamics that are predictive but not attractor-based. Our framework makes these alternatives empirically testable by separating "classification success" from "dynamical success" under controlled regimes.

Limitations and future work. This work focuses on small synthetic tasks (digits and successor modulo 10) and two hand-designed manifolds. Future work could test whether the same shortcut-versus-attractor dichotomy appears under different learning rules, noise models, larger vocabularies, or richer multi-step transitions; whether coupling multiple manifolds (or adding explicit displacement-like inputs) can bridge discontinuities; and whether hierarchical or multi-area architectures can transform non-local jumps into sequences of local moves. More broadly, the evaluation strategy used here suggests a general recipe for studying learned dynamics: explicitly score the behavior you care about (late stability), not the proxy that is easiest to satisfy (early prediction).

5 Conclusion

We introduced an experimental framework for training continuous attractor networks to perform successor-like transitions without externally provided displacement signals, and for disentangling predictive success from attractor-consistent dynamics. We found that short-horizon regimes permit impulse-driven shortcut solutions that achieve high apparent accuracy yet fail to sustain activity during free-run. To obtain meaningful comparisons, model selection must explicitly suppress early transients and evaluate late-trial stability.

Under attractor-enforced evaluation, a Ring topology supports robust learned successor transitions over long horizons, while a folded Snake topology exhibits systematic failures at manifold discontinuities, revealing a geometric limit that simple BG-inspired timing control does not overcome. Together, these results suggest that stable learned transitions are possible but not automatic: they emerge only under constrained evaluation regimes, and their achievable accuracy is strongly shaped by manifold geometry.

Code Availability

An open-source reference implementation is available at `https://github.com/javadan/can-paper`.

References

[1] Shun-ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[2] Kechen Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble. *Journal of Neuroscience*, 16(6):2112–2126, 1996.

[3] Yoram Burak and Ila R. Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS Computational Biology*, 5(2):e1000291, 2009.

[4] Mikail Khona and Ila R. Fiete. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 2022.

[5] H. Sebastian Seung. How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences*, 93:13339–13344, 1996.

[6] William E. Skaggs, James J. Knierim, Harish S. Kudrimoti, and Bruce L. McNaughton. A model of the neural basis of the rat's sense of direction. *Advances in Neural Information Processing Systems*, 1995.

[7] Bruce L. McNaughton, Francesca P. Battaglia, Ole Jensen, Edvard I. Moser, and May-Britt Moser. Path integration and the neural basis of the cognitive map. *Nature Reviews Neuroscience*, 7:663–678, 2006.

[8] Jeff Hawkins, Matthew Lewis, Max Klukas, Scott Purdy, and Subutai Ahmad. A framework for intelligence and cortical function based on grid cells in the neocortex. *Frontiers in Neural Circuits*, 2019.

[9] Jeff Hawkins. *The Thousand Brains Theory of Intelligence*. 2021.

[10] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5:613–624, 1993.

[11] Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20:1643–1653, 2017.

[12] Samuel J. Gershman. The successor representation: Its computational logic and neural substrates. *Journal of Neuroscience*, 38(33):7193–7200, 2018.

[13] C. C. Alan Fung, K. Y. Michael Wong, and Si Wu. A moving bump in a continuous manifold: A comprehensive study of the tracking dynamics of continuous attractor neural networks. *Neural Computation*, 22:752–792, 2010.

[14] 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*, 2020.

[15] Michael J. Frank. Dynamic dopamine modulation in the basal ganglia: A neurocomputational account of cognitive deficits in parkinson's disease. *Journal of Cognitive Neuroscience*, 16:1120–1134, 2004.

[16] Randall C. O'Reilly and Michael J. Frank. Making working memory work: A computational model of learning in the frontal cortex and basal ganglia. *Neural Computation*, 18:283–328, 2006.

[17] Thomas E. Hazy, Michael J. Frank, and Randall C. O'Reilly. Towards an executive without a homunculus: Computational models of the prefrontal cortex/basal ganglia system. *Philosophical Transactions of the Royal Society B*, 2007.

[18] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. *Proceedings of ICML*, 2009.