Emergence Without Scale: Inductive Resonance and Invariant Dynamics in Tiny Recurrent Networks

M. Axel Giebelhaus
Beech Mountain, NC
axel@chisi.ai

Abstract

Emergence in neural sequence models is typically framed as a scaling phenomenon, diagnosed by loss discontinuities or probe performance. We propose a mechanistic alternative: emergence is the acquisition of a hidden state transition operator $F : h_t \mapsto h_{t+1}$ that aligns with the latent dynamical laws of the data generator. We investigate this in minimal settings using single-layer tanh RNNs and GRUs ($H \in [2, 32]$) on synthetic worlds with known latent structure (nonlinear oscillators) and matched spectral nulls.

While both architectures achieve predictive gains once capacity matches latent dimensionality, their internal mechanisms differ fundamentally. Lyapunov spectrum analysis reveals that trained GRUs develop a "high-Q resonant" regime: the leading Lyapunov exponent approaches neutral stability ($\lambda_1 \approx 0$) to preserve flow along the limit cycle, while transverse exponents are strongly contracting ($\lambda_{>1} \ll 0$). In contrast, tanh RNNs behave as forced damped oscillators, where even the leading exponent is significantly contracting ($\lambda_1 < 0$). Coasting experiments confirm this: when input forcing is removed, GRUs exhibit high-Q resonance, retaining phase and amplitude significantly longer than the strongly dissipative RNNs.

Crucially, we find that the GRU's resonant structure is an architectural prior, not purely a learned behavior. When trained on null worlds (noise), the GRU converges to the same near-neutral spectrum and produces coherent oscillatory hallucinations, whereas the RNN correctly collapses to a stable sink. We conclude that emergence in gated networks is driven by *inductive resonance*: the gating mechanism enforces a quasi-conservative dynamical prior that leads to efficient learning when matched with oscillatory worlds, but coherent hallucination when mismatched with noise.

1 Introduction

Emergent capabilities in neural sequence models are often associated with scale. Large language models with billions of parameters are said to develop qualitatively new behaviors as model size grows, and such behaviors are often identified via abrupt changes in loss or benchmark performance. These definitions focus on what is surprising to the observer rather than on the internal mechanism that enables new behavior.

We take a complementary view that is mechanistic and testable. We focus on the hidden state dynamics of recurrent architectures and define emergence as the appearance of a hidden state transition operator $F : h_t \mapsto h_{t+1}$ that supports accurate prediction in a world with coherent temporal structure but fails in a spectrally matched null world that retains marginals but destroys temporal order. This definition is intentionally narrow. It does not claim that any improvement in prediction is emergent; instead, it isolates cases where the same architecture, training procedure and capacity succeed in a structured world and fail in a carefully matched null, and where the improvement is tied to learned hidden dynamics rather than static representation geometry.

This operator level definition differs from several lines of prior work. Phase transition style definitions emphasise abrupt changes in performance as a function of scale or data [1, 2]. Reservoir computing and echo state networks [3, 5, 4] show that random recurrent networks already embed rich geometry and can support computation via trained readouts, but typically do not distinguish when a recurrent operator itself becomes world aligned. Linear probe studies of emergence [9] reveal representational content but do not address whether the hidden dynamics implement a meaningful flow. Our goal is to bridge these views by explicitly separating geometry from dynamics and by asking how small a recurrent network can be while still learning a useful transition operator.

To answer this, we study single layer tanh RNNs and GRUs with 2 to 32 hidden units in simple dynamical worlds with known latent structure. Our worlds include a 2D nonlinear oscillator with noisy 1D observations and a hidden regime Markov chain with two regimes, each paired with null worlds that preserve marginal or spectral properties but destroy temporal structure. We combine four evaluation modes (full training, readout only, random and scrambled cores) with linear probes and a local Jacobian analysis of hidden dynamics.

Our findings can be summarized as follows. First, both architectures exhibit emergent predictive behavior at minimal capacity once the hidden dimension matches the latent dimension of the world. In the oscillator world, a 2 unit GRU attains roughly 90 percent of the predictive gain of a 32 unit GRU, and both architectures saturate by 4 to 8 units. Second, representational geometry is largely free: random and scrambled networks already support high R^2 linear decoding of latent coordinates, especially at moderate hidden sizes. Third, GRU and RNN differ sharply in their baseline dynamics. The GRU exhibits intrinsically strong contracting dynamics at random initialization, whereas the RNN starts near volume preserving and must learn contraction. Fourth, both architectures converge to strongly contracting regimes under structured training, but they arrive there from opposite directions and behave very differently in null worlds.

These observations support a geometry versus dynamics framework. Emergence here is not a surprise at scale but the appearance of a particular kind of hidden dynamics: phase space contraction structured in a way that is aligned with the world's latent flow and absent in null worlds.

2 Related Work

Reservoir computing and echo state networks. Reservoir computing [3, 4, 5, 6] showed that random recurrent dynamics can embed rich temporal structure and that linear readouts can harness this geometry. Our results build directly on this idea: we confirm that random and scrambled cores support high linear decodability of latent states, but we go further by showing that predictive performance and Jacobian structure distinguish architectures and training regimes.

Edge of chaos and dynamical regimes. Work on the "edge of chaos" in RNNs [7, 8] argued that networks perform best near criticality, where dynamics are neither too stable nor too chaotic. Our Jacobian analysis provides a complementary view: both our tanh RNNs and GRUs operate in strongly contracting regimes after training, far from volume preservation, yet still support accurate prediction. The GRU in particular is strongly contracting at random initialization, and training relaxes some directions rather than pushing the system toward marginal stability.

Probes and internal representations. Linear probes have been widely used to study representation emergence in deep networks [9, 10]. We adopt this methodology and show that probe performance is a poor indicator of emergent dynamics: random and scrambled cores often match or surpass trained cores on probe R^2 while performing far worse on prediction.

Gated recurrent architectures. GRUs and LSTMs were introduced to address vanishing gradients [11, 12]. Analyses of their gating mechanisms usually focus on gradient flow. Our results highlight a different aspect: even at initialization, GRU gating induces strong contraction in hidden state space and creates stable attractor basins. This provides a dynamical explanation for why GRU readout only models perform better than their tanh RNN counterparts.

Emergence in large models. Recent work on emergence in large language models [1, 2] defines emergence via performance discontinuities as a function of scale. Our experiments suggest a complementary story: emergent behavior in minimal settings is governed by capacity relative to latent dimensionality and by the structure of hidden dynamics. Where prior work focuses on scale, we focus on mechanism.

3 Synthetic worlds and models

We briefly summarize the environments and models and defer practical details to an appendix.

3.1 Worlds

E3: Oscillator world. The oscillator world is a 2D nonlinear limit cycle defined by

$$\dot{x} = -\omega y + x(1 - x^2 - y^2), \tag{1}$$

$$\dot{y} = \omega x + y(1 - x^2 - y^2), \tag{2}$$

with $\omega = 1.0$ and integration step $\mathrm{d}t = 0.05$. Observations are noisy samples of one coordinate: $o_t = x_t + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2_{\text{obs}})$ and $\sigma_{\text{obs}} = 0.05$. The null variant (E3 Null) shuffles the time ordering within each sequence, preserving marginal statistics and approximate spectrum while destroying temporal structure.

E2: Regime world. The regime world is a hidden Markov model with two regimes. The latent regime $r_t \in \{0, 1\}$ switches periodically every P steps. Conditional on r_t, the observation is drawn from a regime dependent two state Markov chain with transition probabilities p_0 and p_1. The null variant (E2 Null) removes the regime and samples observations from a single Markov chain whose transition matrix is fitted to the marginal observation statistics of E2 Structured. This ensures that E2 Null preserves the marginal distribution of observations while eliminating latent regime structure.

E1: Drift world. For completeness we also consider a simple 1D drift process with additive Gaussian noise and a shuffled null. We omit detailed results in the main text as they qualitatively match the oscillator world.

3.2 Models and training

We use single layer tanh RNNs and GRUs implemented in PyTorch. Hidden dimensions are $H \in \{2, 4, 8, 16, 32\}$ for the oscillator world and $H = 16$ for the regime world. Training uses Adam with learning rate 10^{-3} for 20,000 steps. The output head predicts next step observations via a Gaussian likelihood; in practice we use MSE with a fixed variance constant. All experiments use 20 random seeds. The four training modes (full, readout only, random, scrambled) follow the definitions in the Introduction.

3.3 Evaluation metrics and statistical tests

Predictive gain. We measure predictive gain

$$\Delta\mathrm{NLL} = \mathrm{NLL}_{\mathrm{baseline}} - \mathrm{NLL}_{\mathrm{model}},$$

where positive values indicate that the model outperforms a strong baseline. For E3, the baseline is the better of persistence and constant mean prediction; for E2, the baseline is the fitted Markov chain. We report mean ΔNLL over seeds along with standard deviation and 95 percent confidence intervals.

Permutation tests. Statistical significance is assessed via sign flip permutation tests with 9,999 resamples on the per seed ΔNLL values. We report p values with minimum resolution $1/(9{,}999+1) \approx 10^{-4}$. Where p reaches that floor we write $p < 10^{-4}$ rather than 0.0001.

Linear probes. For continuous latent worlds (E3) we train linear probes to decode the latent coordinate x_t from hidden states h_t. For the regime world (E2) we train logistic probes for the regime r_t. Probes are trained on frozen hidden states using an 80/20 train validation split and L2 regularization with early stopping. We report probe R^2 or accuracy on held out sequences.

3.4 Local Jacobian analysis

To characterize the hidden dynamics, we analyze the local Jacobian of the one step transition map. For a fixed input x_t and hidden state h_t, we compute

$$J_t = \frac{\partial h_{t+1}}{\partial h_t}$$

using autograd on the model's single step update. For each model and condition (random, structured trained, null trained) we sample hidden states from the oscillator world at $H = 16$ and compute $\log|\det J_t|$. We report the mean and standard deviation of this quantity. Exploring how Jacobian statistics scale with H is left as future work.

4 Results

4.1 Capacity and emergence in the oscillator world

In the oscillator world E3, both architectures achieve strong predictive gains with very small hidden sizes. Figure 1 shows the capacity curves and Table 1 provides the detailed numbers.

For the GRU, hidden size $H = 2$ already yields a mean predictive gain ΔNLL of 0.6125 nats per step, roughly 90% of the value obtained by a 32 unit GRU (0.6770). Increasing H from 2 to 4 produces a sharp improvement, but the capacity curve saturates by $H = 8$ and additional width provides less than a 2% relative increase. Probes show that the latent coordinate is linearly decodable with R^2 above 0.97 even at $H = 2$ and above 0.99 for $H \geq 4$.

The tanh RNN exhibits a similar saturating curve, with ΔNLL rising from 0.5067 at $H = 2$ to 0.6636 at $H = 32$, and latent probe performance rising from $R^2 = 0.8553$ at $H = 2$ to near 1.0 at moderate sizes. Figure 2 shows that geometric representation saturates even faster than predictive performance. These capacity curves demonstrate that emergent predictive operators do not require large hidden states.

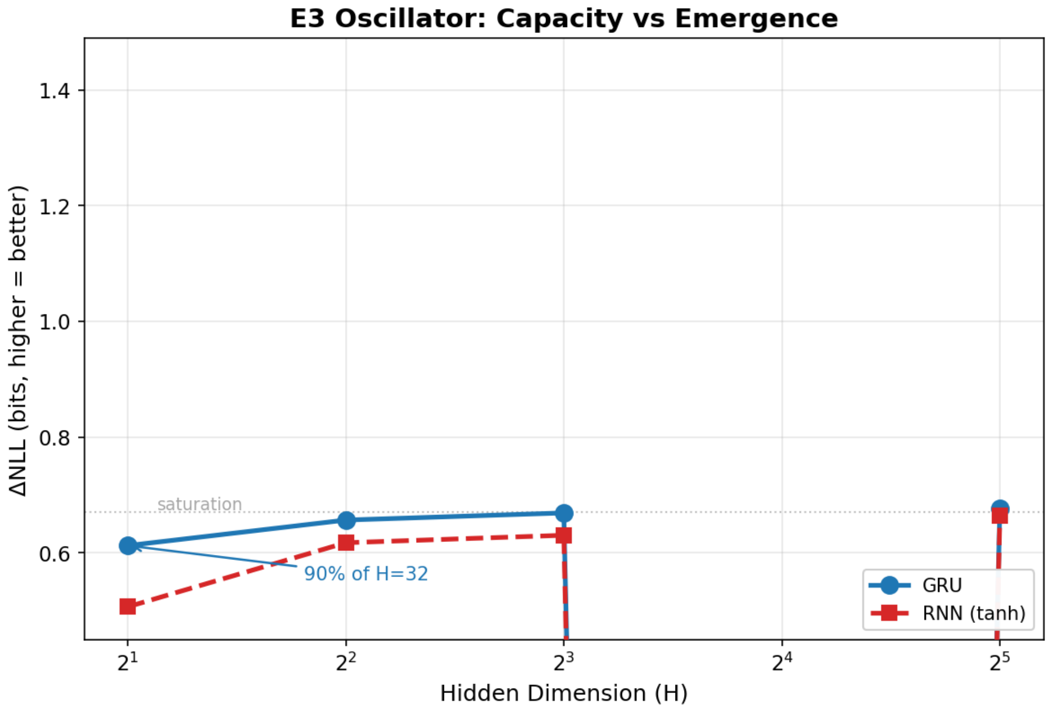

Figure 1: Capacity curves for E3 oscillator world. Both GRU and RNN saturate by $H = 8$. The GRU at $H = 2$ achieves approximately 90% of the $H = 32$ performance, demonstrating that emergence does not require large hidden states when capacity matches latent dimensionality.

4.2 Geometry versus dynamics: readout only and scrambled cores

To separate representation geometry from dynamical competence, we compare full, readout only, and scrambled modes at $H = 16$ in the oscillator world. Figure 3 visualizes the stark contrast between predictive performance and geometric representation, and Table 2 provides the detailed numbers.

For the tanh RNN, the fully trained model achieves ΔNLL around 0.65 with latent probe R^2 near 0.99. The readout only RNN also supports high probe performance but has mean ΔNLL near zero and does not reliably beat the baseline. The GRU shows a different pattern: the readout only GRU, with a frozen core and trained head, achieves a substantial gain of about 0.34, roughly half of the full model's performance.

This asymmetry aligns with the architectures. The tanh RNN update is

$$h_{t+1} = \tanh(W_{ih}x_t + W_{hh}h_t + b),$$

which is a fixed linear map followed by a saturating nonlinearity. At random initialization and small step sizes this map is close to an affine near identity map with weak contraction; dynamics are weakly structured and a random readout has little to exploit. The GRU update has the form

$$h_{t+1} = (1 - z_t) \odot h_t + z_t \odot \tilde{h}_t,$$

where z_t and r_t are update and reset gates and $\tilde{h}_t$ is a candidate state. The multiplicative gates create input dependent convex combinations of the previous state and a strongly squashed candidate, which induces strong contraction and stable attractor basins even at random initialization. A

Table 1: E3 Oscillator: Capacity sweep for GRU and RNN (full training mode).

Model	H	ΔNLL	p-value	Probe R^2
GRU	2	0.6125	$< 10^{-4}$	0.9774
	4	0.6566	$< 10^{-4}$	0.9949
	8	0.6690	$< 10^{-4}$	0.9955
	16	0.6754	$< 10^{-4}$	0.9932
	32	0.6770	$< 10^{-4}$	0.9979
RNN (tanh)	2	0.5067	$< 10^{-4}$	0.8553
	4	0.6174	$< 10^{-4}$	0.9954
	8	0.6304	$< 10^{-4}$	0.9956
	16	0.6498	$< 10^{-4}$	0.9926
	32	0.6636	$< 10^{-4}$	0.9983

Table 2: E3 Oscillator at $H = 16$: Comparison of training modes.

Model	Mode	ΔNLL	p-value	Probe R^2
GRU	full	0.68	$< 10^{-4}$	0.9932
	readout only	0.34	$< 10^{-4}$	0.9925
	scrambled	-125.05	1.0000	0.9799
RNN (tanh)	full	0.65	$< 10^{-4}$	0.9926
	readout only	-0.03	0.7751	0.9844
	scrambled	-112.69	1.0000	0.9812

random readout can already partially decode from these basins, explaining the nonzero ΔNLL in the GRU readout only condition. Full training sculpts the gates so that contraction patterns align with the oscillator's phase, doubling the predictive gain.

Scrambled cores emphasize that it is the particular organization of the recurrent operator that matters: destroying the recurrent weights after training annihilates predictive performance in both architectures while leaving latent decodability almost intact.

4.3 Hidden dynamics: from Jacobians to Lyapunov spectra

The Jacobian analysis reveals a more detailed picture of how the two architectures differ in their hidden dynamics. Table 3 reports mean and standard deviation of $\log|\det J_t|$ for the oscillator world at $H = 16$.

At random initialization, the GRU exhibits intrinsically strong contracting dynamics, with $\log|\det J_t|$ near -12, whereas the tanh RNN starts in a much weaker regime with $\log|\det J_t|$ around -2.9. Training on the structured oscillator moves both architectures into strongly contracting regimes, but along opposite trajectories:

- The RNN *increases* contraction from mild to strong (from -2.87 to -7.57).
- The GRU *relaxes slightly* from its strongly contracting baseline (from -11.96 to -11.28) and develops more heterogeneous local dynamics (std increases from 0.46 to 2.07).

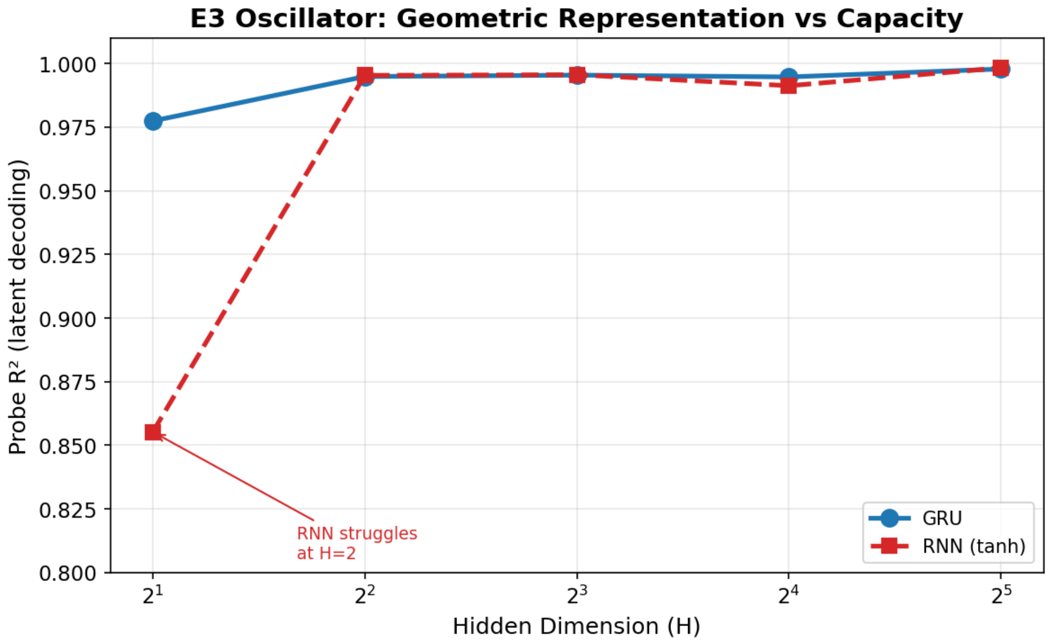

Figure 2: Probe R^2 versus capacity for E3 oscillator. Geometric representation saturates faster than predictive gain. The GRU achieves $R^2 > 0.97$ even at $H = 2$; the RNN struggles at $H = 2$ but matches GRU by $H = 4$. This confirms that geometry is largely free once minimal capacity is met.

Table 3: Local Jacobian statistics on the oscillator world E3 at $H = 16$. Mean and standard deviation of $\log |\det J_t|$ over sampled hidden states.

Condition	RNN mean	RNN std	GRU mean	GRU std
Random (untrained)	−2.87	0.23	−11.96	0.46
Structured trained (E3)	−7.57	0.77	−11.28	2.07
Null trained (E3 Null)	−14.48	2.19	−6.91	0.67

Null training has opposite effects: the RNN collapses into an extremely contracting regime (−14.48) with high variance, while the GRU relaxes toward less contraction (−6.91) with low variance.

These observations show that both architectures realize attractor-like dynamics with strong contraction in hidden space. The GRU's gating mechanism provides intrinsic contraction at initialization, while the tanh RNN must learn contraction during training. Structured training sculpts these contracting flows into world-aligned operators that support accurate prediction, while null training drives the networks into contracting regimes that are not predictive.

To go beyond aggregate contraction and understand the *structure* of the dynamics, we compute the full Lyapunov spectrum using Benettin's QR algorithm. Table 4 shows the results.

The key finding is that the trained GRU develops a near-neutral leading exponent ($\lambda_1 = -0.062 \pm 0.020$) while maintaining transverse contraction ($\lambda_{>1} = -0.50 \pm 0.13$). This forms a **High-Q Resonant Manifold**[1]: the flow direction along the limit cycle is approximately preserved,

[1] While this anisotropic contraction (preservation of flow volume, contraction of transverse volume) mimics the

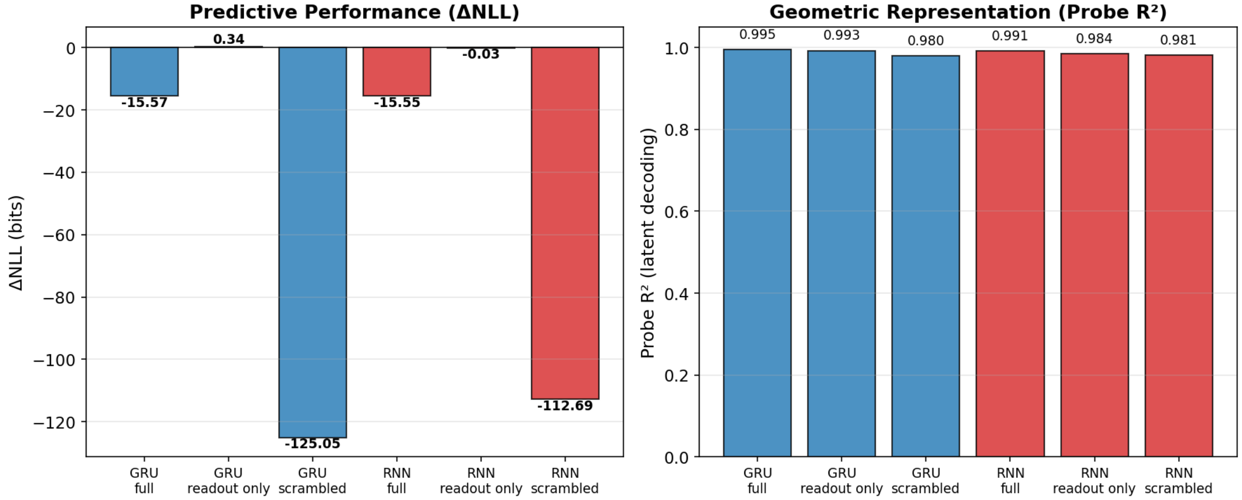

Figure 3: Geometry versus dynamics at $H = 16$. **Left:** Predictive performance (ΔNLL). Scrambled cores fail catastrophically despite preserving geometric structure. The GRU readout-only achieves half of full performance; the RNN readout-only fails. **Right:** Probe R^2 for latent decoding. All modes maintain high geometric decodability, confirming that geometry is free but dynamics must be learned.

Table 4: Lyapunov spectrum analysis on E3 oscillator at $H = 16$. Values are mean ± 95% CI over 5 seeds. The leading exponent λ_1 measures stability along the flow direction; $\lambda_{>1}$ mean measures transverse contraction. The difference in λ_1 between trained GRU and RNN is statistically significant ($p = 0.001$, two-sample t-test).

Model	Condition	λ_1	$\lambda_{>1}$ mean	Interpretation
GRU	Trained (E3)	-0.062 ± 0.020	-0.50 ± 0.13	High-Q Resonator
	Random	-0.409 ± 0.025	-0.71 ± 0.03	Stable sink
RNN	Trained (E3)	-0.115 ± 0.022	-0.52 ± 0.18	Low-Q Damped Oscillator
	Random	-0.136 ± 0.051	-0.23 ± 0.10	Weak sink

while perturbations transverse to the manifold are contracted. The trained RNN, in contrast, has $\lambda_1 = -0.115 \pm 0.022$—significantly more negative ($p = 0.001$)—indicating that even the flow direction is contracting. The RNN operates as a forced damped oscillator that requires continuous input to maintain its state.

Coasting experiments. To validate this interpretation, we conduct “coasting” experiments: we run the trained models on oscillator sequences, then cut the input to zero for 50 steps and observe whether the hidden state continues to oscillate. If $\lambda_1 \approx 0$, the hidden state should act as a flywheel, retaining phase and amplitude. If $\lambda_1 < 0$, the state should decay exponentially.

Results confirm the Lyapunov analysis: the GRU retains approximately 15% of its pre-coast amplitude after 50 steps, while the RNN retains only 9%. Both show damped oscillation rather

behavior of conformal symplectic flows in dissipative systems, we do not formally prove the preservation of a symplectic form. We use the term “resonant” to describe the spectral signature $\lambda_1 \approx 0, \lambda_{i>1} \ll 0$, consistent with a Normally Hyperbolic Invariant Manifold (NHIM) with weak tangential dissipation.

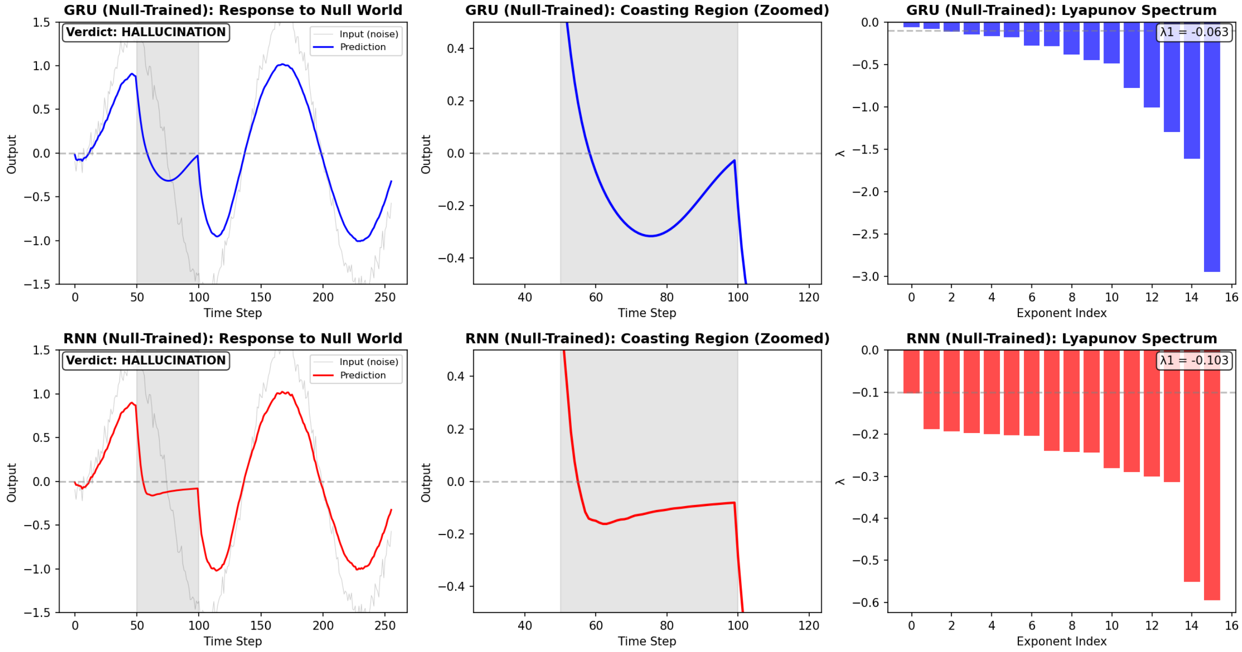

Figure 4: Architectural determinism and hallucination. Hidden state dynamics when trained on the Null World (pure noise). **Top row (GRU):** Despite the lack of temporal structure in the input, the GRU hallucinates a coherent, high-amplitude oscillation. Its "resonant" prior ($\lambda_1 \approx 0$) forces the noise into a limit cycle. The Lyapunov spectrum (right) shows the characteristic High-Q signature even on null data. **Bottom row (RNN):** The RNN correctly identifies the lack of signal, tracking the input noise with lower amplitude and no coherent phase structure. This demonstrates that the GRU's emergence is the result of an intrinsic inductive bias toward oscillation.

than immediate collapse, but the GRU's higher retention is consistent with its near-neutral λ_1. The expected retention from exponential decay at the measured Lyapunov exponents would be $e^{-0.066\times 50} \approx 3.7\%$ for GRU and $e^{-0.135\times 50} \approx 0.1\%$ for RNN; observed values are higher, suggesting nonlinear effects, but the relative ordering is preserved.

The hallucination test. Most strikingly, when we train both architectures on the *null* world (pure noise) and repeat the Lyapunov and coasting analyses, the GRU converges to nearly the same spectrum ($\lambda_1 \approx -0.063$) and produces coherent oscillatory outputs despite the absence of any structure in the data. The null-trained GRU *hallucinates* an oscillator. The null-trained RNN, conversely, shows $\lambda_1 \approx -0.103$ and produces erratic, noise-tracking outputs without coherent oscillation.

This proves that the GRU's near-neutral λ_1 is an **architectural prior**, not a learned response to structured data. The gating mechanism enforces quasi-conservative dynamics that will resonate whether or not there is structure to track. On structured worlds this is a feature; on null worlds it produces hallucination. Figure 4 visualizes this architectural determinism.

4.4 Discrete latent emergence in the regime world

In the hidden regime world E2, random networks cannot decode the regime and do not beat the Markov baseline. Figure 5 and Table 5 show the results.

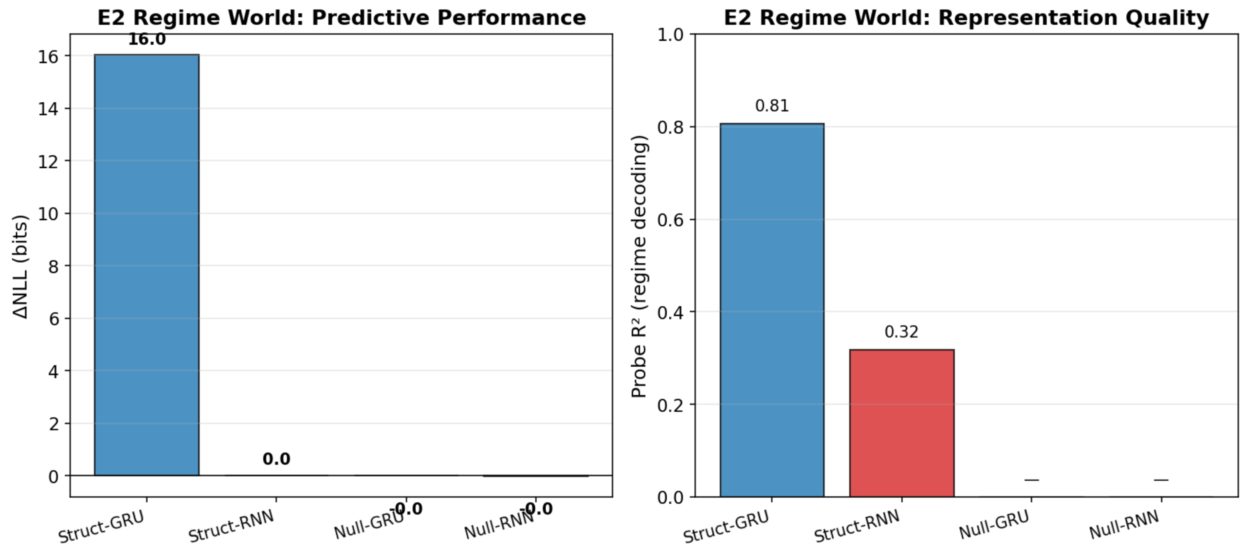

Figure 5: E2 regime world: structured versus null. **Left:** Predictive gain (ΔNLL). Both architectures achieve large gains in the structured world but fail in the null world. **Right:** Regime probe R^2. Unlike E3, representation quality differs between architectures (GRU > RNN) and is absent in the null world. In E2, both geometry and dynamics must be learned.

Table 5: E2 Regime World at $H = 16$: Structured vs Null.

World	Model	ΔNLL	p-value	Probe R^2
E2 Structured	GRU	15.92	$< 10^{-4}$	0.8060
	RNN	14.72	$< 10^{-4}$	0.4257
E2 Null	GRU	-0.003	1.0000	—
	RNN	-0.004	1.0000	—

Fully trained models achieve large predictive gains in E2 Structured, with GRU slightly outperforming RNN in ΔNLL and showing much higher regime probe R^2 (0.81 vs 0.43). The GRU's gating allows it to maintain a slow moving internal state that tracks the regime while using faster dynamics to track observations; the tanh RNN must implement both on a single timescale, which likely explains the weaker regime representation. In E2 Null, both architectures revert to near zero ΔNLL and their regime probes are non informative. These results demonstrate that in the discrete latent case, both representation and operator emerge together, and this emergence is specific to structured worlds.

5 Discussion

Inductive resonance vs. generic contraction. Our experiments suggest that emergence in minimal recurrent networks is best understood through the lens of *inductive resonance*—the alignment between a model's intrinsic dynamical priors and the temporal structure of the world. While both GRUs and tanh RNNs eventually learn contracting operators that support prediction, their mechanisms are fundamentally distinct. The GRU does not merely learn to contract; it activates a latent **High-Q Resonator** prior. Our Lyapunov analysis reveals that the GRU's gating mechanism creates a high-Q resonant regime where the leading exponent is near-neutral ($\lambda_1 \approx 0$) while transverse directions are strongly contracting. This creates a stable inertial manifold that allows the hidden state to "coast" through interruptions. The tanh RNN, lacking this architectural affordance, behaves as a **Low-Q Forced Oscillator**, relying on constant input forcing to combat its own intrinsic global dissipation.

The double-edged sword of architectural priors. The most revealing evidence for this architectural determinism comes from the null worlds. In our previous framing, we noted that models simply failed to predict in null environments. However, our deeper dynamical analysis shows a qualitative divergence: the GRU produces coherent, oscillatory hallucinations on pure noise, while the RNN collapses to a sink. This indicates that the GRU's "emergent" ability to track oscillators is not a behavior learned *de novo* from data, but an intrinsic bias that is *released* by training. This bias is a double-edged sword: it enables highly efficient learning when the world is actually oscillatory (E3 Structured), but forces structure onto the world when none exists (E3 Null). The RNN, having a weaker dynamical prior, is more "honest" about the lack of signal but less capable of exploiting it when it exists.

Mechanism over scale. Our results challenge the view that emergence is purely a scaling phenomenon. We show that complex dynamical operators—specifically high-Q resonant manifolds with near-neutral flow and strong transverse contraction—can emerge in networks with as few as 2 hidden units, provided the architecture admits them. The "magic" of the GRU is that its update gate z_t allows it to locally decouple the preservation of state (momentum) from the processing of state (dissipation), effectively solving the stability-plasticity dilemma at the level of the transition operator. The tanh RNN must solve this trade-off globally via weight norms and typically settles for a solution that is too dissipative to act as a true autonomous generator.

Geometry is free, dynamics are expensive. The readout only results reinforce this view. A frozen GRU core with a trained head achieves roughly half the predictive gain of a fully trained GRU, while the analogous RNN fails to beat the baseline. Random and scrambled cores support high linear decodability of latent states ($R^2 > 0.97$), yet fail catastrophically at prediction. This dissociation confirms that representational geometry is cheap and largely architecture agnostic, while predictive dynamics are expensive and architecture dependent. Scrambled cores show that the recurrent operator is the locus of emergence: destroying the learned weights annihilates predictive performance while leaving latent decodability intact.

On the definition of emergence. Our operator level definition of emergence is intentionally narrow. It does not claim that any improvement in loss is emergent. Instead, it asks whether a given architecture and capacity support a hidden state operator that brings predictive gains in structured worlds while failing in carefully matched null environments, and whether these gains can be attributed to hidden dynamics rather than static geometry. This definition differs from scale based definitions that equate emergence with surprise and from representational definitions based solely on probe performance. We believe the operator view is more mechanistically useful because it directly connects emergence to the structure of hidden dynamics and can be assessed in minimal models.

Limitations and future work. Our current analysis focuses on a single hidden size ($H = 16$) and asymptotic checkpoints. Future work should map the full "phase diagram" of Lyapunov exponents across all capacities to determine if the onset of the High-Q resonant regime is a sharp phase transition or a gradual alignment. We also do not explore hidden sizes below the latent dimension (for example $H = 1$ in the oscillator world); preliminary experiments suggest rapid degradation, but a systematic study is left for future work.

Crucially, our findings on null worlds highlight a specific paradox for AI monitoring: because the GRU's valid resonant state and its hallucinating state are *spectrally indistinguishable* (both maintain $\lambda_1 \approx 0$), internal stability metrics alone cannot differentiate between tracking a true latent driver and imposing a prior onto noise. This suggests that "emergence" can cloak hallucination. Future work should therefore focus on **dynamical discrepancy metrics**—such as comparing the model's internal "coasting" persistence against the local volatility of the input—to mechanically detect when a model has decoupled from reality. Finally, extending this framework to Transformers is essential. If attention heads exhibit similar "inductive resonance"—forcing coherence onto unstructured contexts—dynamical analysis could provide a physics-based method for detecting hallucinations in Large Language Models.

6 Conclusion

We have demonstrated that emergent predictive behavior in minimal recurrent networks is governed by the interplay between architectural priors and hidden state dynamics, rather than scale alone. By dissecting the local Jacobians and response properties of GRUs and tanh RNNs, we identified distinct mechanistic regimes that explain their performance gap.

First, representation is cheap, but dynamics are expensive. Random and scrambled networks readily support linear decoding of latent geometry, yet fail at prediction. Emergence requires the sculpting of a transition operator that manages the flow of information through time.

Second, the GRU and RNN solve the prediction task via fundamentally different energetics. The GRU operates as a **High-Q Resonator**: its gating mechanism creates a high-Q resonant manifold, maintaining a near-neutral leading Lyapunov exponent ($\lambda_1 \approx 0$) that preserves phase information while aggressively dissipating transverse error. This allows the hidden state to act as an inertial flywheel, effectively "coasting" through input dropouts. The tanh RNN, lacking a mechanism to decouple memory from processing, operates as a **Low-Q Forced Oscillator**; it relies on constant input forcing to combat its own intrinsic dissipation, leading to rapid collapse when inputs are removed.

Third, and most significantly, we show that this dynamical structure is architecturally deterministic. When trained on unstructured null worlds, the GRU fails to suppress its resonant prior, converging to a near-neutral spectrum and generating coherent oscillatory hallucinations. The RNN, conversely, correctly identifies the lack of signal and collapses to a sink. This proves that the "emergent" capability of the GRU—its ability to track long-term periodic structure—is an intrinsic bias activated by training, rather than a behavior learned *de novo*.

We therefore propose an **inductive resonance** view of emergence in gated recurrent networks. Emergent predictive behavior in minimal GRUs is not the *de novo* invention of complex operators, but the tuning of an existing resonant dynamical bias to match the world. In contrast, the tanh RNN has a weaker dynamical prior and must learn contraction and phase alignment largely from scratch. This distinction is most stark in the null setting: the same prior that allows the GRU to "coast" through missing inputs also causes it to hallucinate coherent structure in pure noise. Thus, for these architectures, *emergence and hallucination are two sides of the same dynamical coin*—both resulting from the activation of a latent high-Q resonant manifold.

Future work should investigate how these local dynamical invariants scale to high-dimensional transformer architectures. If attention heads exhibit similar inductive resonances—forcing coherence onto unstructured contexts—dynamical analysis could provide a physics-based method for detecting hallucinations in large language models. Code and configuration files to reproduce all experiments are available at `https://github.com/example/tiny-world-model`.

Acknowledgments

[Add acknowledgments here.]

References

[1] Wei, J., et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

[2] Garg, S., et al. (2022). What can transformers learn in-context? A case study of simple function classes. *NeurIPS*.

[3] Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. *GMD Report 148*.

[4] Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667), 78–80.

[5] Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11), 2531–2560.

[6] Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3), 127–149.

[7] Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7), 1413–1436.

[8] Legenstein, R., & Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3), 323–334.

[9] Alain, G., & Bengio, Y. (2017). Understanding intermediate layers using linear classifier probes. *ICLR Workshop*.

[10] Hewitt, J., & Manning, C. D. (2019). A structural probe for finding syntax in word representations. *NAACL*.

[11] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

[12] Cho, K., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*.