CONSTRAINT BREEDS GENERALIZATION: TEMPORAL DYNAMICS AS AN INDUCTIVE BIAS

Xia Chen
Georg Nemetschek Institute
Munich Data Science Institute
Technische Universität München
x.c.chen@tum.de

ABSTRACT

Conventional deep learning prioritizes unconstrained optimization, yet biological systems operate under strict metabolic constraints. We propose that these physical constraints function not as limitations, but as a temporal inductive bias that breeds generalization. Through a phase-space analysis of signal propagation, we reveal a fundamental asymmetry: expansive dynamics amplify noise, whereas proper dissipative dynamics compress phase space, compelling the abstraction of invariant features. This condition can be imposed externally via input encoding, or intrinsically through the network's own temporal dynamics. Both pathways require architectures capable of temporal integration and proper constraints to decode induced invariants, whereas static architectures fail to capitalize on temporal structure. Through comprehensive evaluations across supervised classification, unsupervised reconstruction, and zero-shot reinforcement learning, we demonstrate that a critical "transition" regime maximizes generalization capability. These findings establish dynamical constraints as a distinct class of inductive bias, suggesting that robust AI development requires not only scaling and removing limitations, but computationally mastering the temporal characteristics that naturally promote generalization.

***K*eywords** Temporal Inductive Bias · Generalization · Neuromorphic Computing · Spiking Neural Networks · Dissipative Dynamics

1 Introduction

Modern deep learning paradigms largely abstract the temporal dimension, simplify time to a spatial index to enable efficient parallelization[1, 2, 3]. This modification contrasts sharply with biological systems, where temporal dynamics are intrinsic to computation and operate under severe energetic constraints[4, 5]. While current AI equates generalization with unconstrained scaling[6, 7], the brain's strict metabolic budget forces neural dynamics into a dissipative regime characterized by sparse, temporally precise activity[8, 9, 10]. We propose that this energy-efficiency trade-off functions not as a limitation, but also as a fundamental *inductive bias*: by naturally compressing the solution space over time, these physical constraints compel the system to abstract robust invariants from noisy sensory streams.

Specifically, we hypothesize that networks trained on temporal-encoded, well-structured inputs will inherently learn representations invariant to task-irrelevant variations, treating time not merely as a dimension for indexing but as a physical resource for computation. Building on recent work in dynamical alignment[11], which demonstrated that SNN computational modes can be sculpted by input dynamics, we posit that temporal constraints serve as an active regularizer. To instantiate this principle computationally, we leverage Spiking Neural Networks (SNNs), which provide the necessary theoretical foundation for temporal processing and biological plausibility[12, 13].

We designed experiments sequentially based on emergent observations and established that constraint, specifically structured dissipative temporal dynamics, breeds generalization across:

1. Representational level: In cross-encoding classification, networks trained under contractive constraints exhibit asymmetric generalization capability while networks trained under expansive dynamics fail (Experiment 1).

2. Structural level: In unsupervised learning, proper contractive dynamics spontaneously induce biologically plausible, structured receptive fields, whereas other regimes yield isotropic noise (Experiment 2).
3. Behavioral level: In reinforcement learning, where inputs are inherently temporal, we validate the constraint principle through two complementary pathways: encoding-level constraints (via dissipation parameter δ) and architecture-level constraints (via membrane leak β). Both pathways achieve superior zero-shot transfer to unseen physical environments, outperforming baseline architectures (Experiment 3).

Furthermore, we discover that this generalization capability requires the co-dependence of temporal input dynamics and processing architecture: time-indexing (static) networks fail to benefit from these dynamical constraints even when provided with identical inputs. Spectral analysis reveals that the proper contractive input trajectory (we called "transition regime") emerges as a "low-frequency, high-entropy" signature, which aligns with the theory of spectral bias in neural networks[14]. These findings suggest dynamical constraints as a distinct class of inductive bias[15] working on the temporal dimension. Unlike explicit, hand-crafted spatial regularizers (e.g., data augmentation[16, 17] or architectural design[18]), this implicit temporal approach operates through a physical phase-space contraction ($\sum \lambda_i < 0$) process that actively induces the network to learn stable, invariant features over time and consistently robust across representational, structural, and behavioral levels.

2 Experiments

2.1 Method: Dynamical Constraints via Dual Pathways

We implement dissipative constraints through two complementary pathways, thereby establishing the framework's generality. Both pathways control the same physical quantity: the rate of phase space contraction during the temporal evolution of signals, measured by the global Lyapunov sum ($\Sigma\lambda_i$)[19].

Encoding-Level Constraints. For tasks with static inputs (Experiments 1-3), we transform features $\mathbf{x} \in \mathbb{R}^d$ into temporal trajectories using a parametrically controllable duffing oscillator system[20]. Each input dimension initializes a three-dimensional dynamical system that evolves for time T with N discrete steps, producing trajectories $\phi_\delta(\mathbf{x}, t) \in \mathbb{R}^{d\times N\times 3}$. The dissipation parameter δ controls the system's phase space behavior:

- *Expansive* ($\delta < 0$): Divergent dynamics ($\Sigma\lambda_i > 0$) that amplify initial conditions.
- *Transition* ($\delta = 0 - 2.0$): Weakly dissipative dynamics ($\Sigma\lambda_i \leq 0$).
- *Dissipative* ($\delta \gg 0$): Contractive dynamics ($\Sigma\lambda_i \ll 0$) that suppress state evolution.

The trajectory's origin is invariant to δ; the parameter solely governs the subsequent temporal evolution of the signal. This design allows us to tune the global phase space dynamics via a single parameter δ within the same system to avoid geometric confounds inherent when comparing distinct chaotic systems. As established in prior work[11], the computational capability of downstream networks is primarily governed by the global phase space contraction ($\Sigma\lambda_i$) of the input, rather than local chaotic divergence (λ_{max}). Therefore, tuning δ enables us to explicitly modulate the degree of constraint imposed on the temporal structure (see Appendix A for detailed dynamical justification).

Architecture-Level Constraints. For tasks with inherently temporal inputs (Experiment 3), we also leverage the SNN's intrinsic dissipation mechanism. The membrane leak parameter β governs temporal integration within the network:

$$\text{mem}_{t+1} = \beta \cdot \text{mem}_t + \text{input}_t$$

Specifically, β governs the effective integration window ($\tau_{\text{mem}} = -1/\ln(\beta)$), acting as the architectural counterpart to the encoding parameter δ. High β (≈ 1.0) preserves information across timesteps (weak constraint), while low β (≈ 0.1) rapidly dissipates internal states (strong constraint). This setup allows Experiment 3 to validate that our findings derive from the constraint principle itself, rather than properties specific to the input encoding.

2.2 Experiment 1: Emergent Generalization in Classification

We first investigate *representational transfer* as a direct proxy for invariance. The rationale is straightforward: If dissipative constraints indeed lead to the abstraction of robust features, a network trained on one dynamical encoding should generalize to others, analogous to domain adaptation but operating within the spectrum of dynamical regimes.

Specifically, we test whether networks trained under different dynamical encodings exhibit generalization across regimes. We designed a cross-encoding evaluation protocol evaluating five networks on the handwritten digits dataset (64 features, 10 classes)[21]: (1) SNNs with Leaky Integrate-and-Fire Neuron (LIF)[13], compared against recurrent

controls: (2) Long Short-Term Memory networks (LSTMs)[22], (3) Recurrent Neural Networks (RNNs); as well as static baselines: (4) Multi-Layer Perceptrons (MLPs) processing only the final timestep (*Last-T*), (5) MLPs processing time-averaged inputs (*Avg-Pool*). All networks share an identical three-layer fully-connected topology to isolate algorithmic differences. We constructed a 12×12 generalization matrix by training each architecture on a specific dynamical regime (δ_{train}) and evaluating it across the full spectrum ($\delta_{\text{test}} \in \{-1.5, \ldots, 10.0\}$). Detailed experiment settings are provided in Appendix A.1.

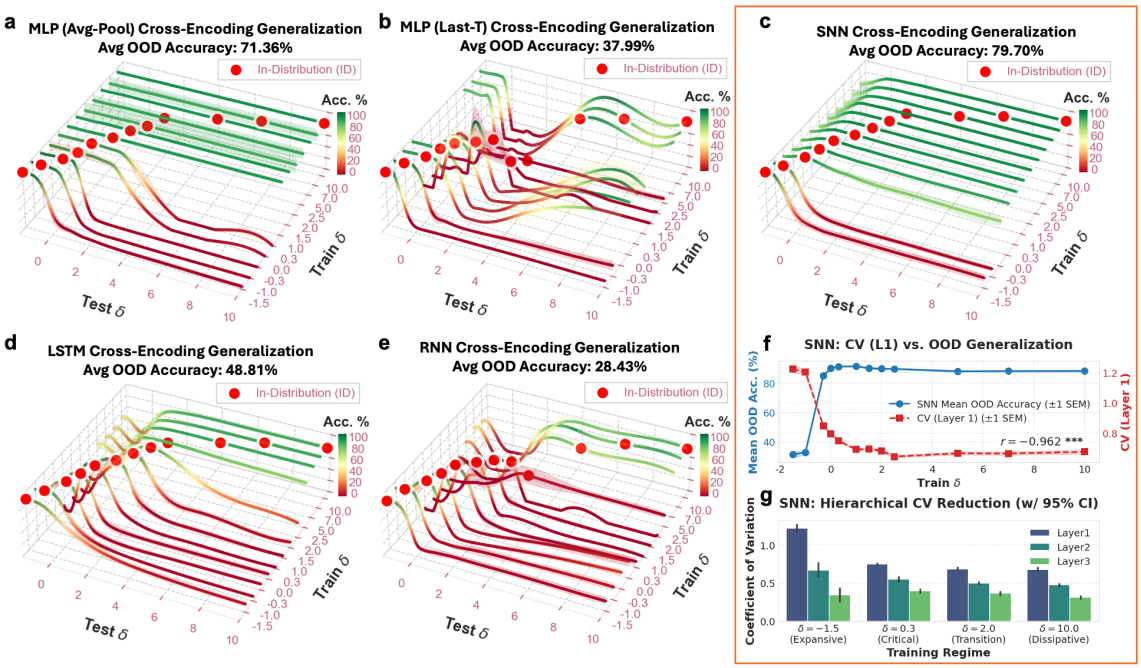

Figure 1: **Cross-encoding generalization with hierarchical variance reduction. (a-e)** 3D generalization landscapes (mean accuracy with deviation, $n = 10$). **(c)** SNNs exhibit a robust generalization in the transition regime ($\delta \approx 0 - 2.0$) across the spectrum. This contrasts with the diagonal-only ridges seen in expansive regimes ($\delta < 0$) and in baseline architectures (*Last-T MLP, LSTM, RNN*; **b, d, e**), and *Avg-Pool MLP* (**a**) shows partial, unstable generalization (see Appendix Tables 6–9). **(f)** Out-of-Distribution accuracy strongly anti-correlates with Layer 1 neural variability (CV; $r = -0.962$). Error bars: ± 1 SEM. **(g)** Hierarchical variance reduction (L1 $\rightarrow$ L3) emerges in SNN across regimes.

Asymmetric Generalization Landscapes. Figure 1 presents the resulting generalization capability as a 3D topographic landscape. We define In-Distribution (ID) accuracy as the performance on the training encoding (the diagonal ridges where $\delta_{\text{train}} \approx \delta_{\text{test}}$), and Out-of-Distribution (OOD) accuracy as the performance across the rest of the spectrum. While all architectures maintain high ID performance (indicated by the peaks of the ridges), SNNs (Figure 1c) reveal a unique, asymmetric generalization capability: Networks trained in expansive regimes ($\delta_{\text{train}} < 0$) form sharp, narrow ridges. They achieve high ID accuracy but collapse immediately in the $\delta_{\text{test}} > 0$ region. This behavior is characteristic of memorization without transfer. In contrast, SNNs trained in the transition ($\delta_{\text{train}} \approx 0$) and dissipative ($\delta_{\text{train}} > 2$) regimes form a robust "generalization plateau." These ridges maintain high altitude across the entire δ_{test} axis. Notably, the transition zone ($\delta_{\text{train}} \approx$ 0-2) achieves the strongest overall generalization, effectively decoding inputs from all other dynamical regimes.

This plateau is absent in recurrent controls (LSTMs, RNNs) and stateless MLPs (*Last-T*), which show OOD failure (collapsed ridges) when tested away from their training diagonal. The *Avg-Pool MLP* shows partial generalization but with significantly higher volatility (see Appendix Tables 5–9). This suggests that the mechanistic shortcut of averaging over the temporal dimension diminishes generalization capacity, as it primarily arises from a particular temporal structure of trajectories decoded by SNN's integral dynamics, rather than from mere temporal aggregation.

Emergent Hierarchical Variance Reduction. *How do SNNs translate these dynamical constraints into robust representations?* We hypothesized that dissipative constraints drive the network to suppress noise and amplify stable features. To quantify this, we measured the neural response stability via the coefficient of variation ($CV = \sigma/\mu$) of mean firing rates across test encodings. Figure 1f reveals a mechanistic insight: neural response variability (CV) strongly anti-correlates with OOD accuracy across the dynamical spectrum ($r = -0.96$ at regime level; robust at sample level with $r = -0.78$, see Appendix Table 10). Furthermore, this stabilization is not uniform but forms hierarchically

over layers. As shown in Figure 1g, SNNs exhibit a monotonic reduction in variability across depth (L1 $\rightarrow$ L3). This emergent phenomenon is highly similar to the processing hierarchy of the biological vision system (primate ventral stream), where raw, stimulus-specific volatility (akin to V1 responses) is progressively distilled into abstract, invariant representations (as in the inferotemporal cortex) [23, 24], while previous works primarily emphasize prior spatial structure design, such as pooling or strided convolutions[25] to achieve invariance.

Another evidence lies in the stability of these representations. Despite high in-distribution performance, a high fluctuating variance observed in the deepest layer (Layer 3) under the expansion regime (0.470 ± 0.288, see Appendix Table 11) suggests that the network memorizes specific dynamical trajectories. In contrast, dissipative constraints drive this final representation toward a tightly converged state (0.307 ± 0.051), indicating the formation of invariant representation. This intrinsic stability reveals a different path: dissipative constraints naturally induce hierarchical stability without architectural specification.

2.3 Experiment 2: Spontaneous Induction of Structured Features

Previous observation from the experiment motivates us for subsequent inquiry: *Does this variance reduction spontaneously induce organized feature representations?* While classical theories posit sparse coding as an explicit optimization objective for cortical receptive fields [9, 26], we hypothesize that dissipative dynamics act as a fundamental constraint that naturally induces the emergence of invariant, structured features. In other words, representational structure is not the result of a designed objective but an emergent property of phase space compression.

To test this, we trained a SNN autoencoder to reconstruct static image patches from temporally encoded inputs. The architecture features a LIF bottleneck layer between a linear encoder (W_{enc}) and decoder (W_{dec}). Critically, the decoder reconstructs from the *temporal sum* of spikes: $\mathbf{x}_{\text{recon}} = W_{\text{dec}} \sum_{t=1}^{T} S(t)$. We followed standard comparative coding schemes[27, 28] and compared eight temporal encoding strategies on CIFAR-10 patches[29], including *Baseline* (static), *Random* (temporal jitter), *Linear* (rate coding), *Poisson* (stochastic spiking), and four dynamical regimes (*Expansive, Critical, Transition, Dissipative*). The network minimized a composite loss balancing reconstruction fidelity and code sparsity. We quantified the organization of learned spatial features at the bottleneck using *RF Standard Deviation* (σ_{RF}) (distinguishing high-contrast structured filters from low-variance noise), alongside *reconstruction* and *sparsity losses*. We validate robustness by varying: (1) sparsity weight λ (specifically including the unconstrained $\lambda = 0$ case) and (2) repeating key experiments on STL-10 (96×96 images, higher resolution)[30]. Detailed experiment protocols are provided in Appendix A.2.

Structure Emergence under Dynamical Constraints Figure 2 provides a qualitative visualization of feature emergence. *Transition dynamics* ($\delta \approx 2.0$) spontaneously produce receptive fields with clear spatial organization: localized, oriented filters resembling the biological orientation maps of the visual cortex. In contrast, *baseline*, *random*, and *expansive* conditions yield homogeneous, unstructured weight distributions resembling salt-and-pepper noise. Table 1 quantifies these distinct organizations across three metrics:

- *Expansive dynamics ($\delta = -1.5$)*: Achieves excellent reconstruction (0.0013) but fails catastrophically at sparsity (1.72), which means that dynamics amplify temporal variations, allowing the network to "memorize" inputs via high-rate firing without learning compressive features.
- *Poisson encoding*: Demonstrates the opposite extreme with near-perfect sparsity (10^{-5}) but minimal structure (0.049). This proves that sparse firing alone is insufficient; temporal stability is required to induce structural differentiation.
- *Transition dynamics ($\delta \approx 2.0$)*: Presents an optimal regime. Its contractive phase space simultaneously reduces activation volume (sparsity 0.0009), suppresses noise to enable clean filters, and amplifies stable features (structure 0.206, $\mathbf{4.9}\times$ higher than alternatives).

Further analysis confirms that even without sparsity penalty ($\lambda = 0$, see Appendix A.2), the *Transition* regime still spontaneously induces structured representations ($\sigma_{\text{RF}} \approx 0.177$), whereas others collapse into noise. In this regime, structure emerges intrinsically by the dynamical constraint, independent of explicit regularization. This offers a principled hypothesis for biological sparse coding: under dissipative metabolic constraints, structured representations emerge not as a pre-programmed objective, but as a canonical solution to learning under temporal constraints[8].

Mechanism This structural emergence raises a fundamental question: *What specific physical property of the transition dynamics drives this outcome?* Spectral analysis confirms that the transition regime ($\delta \approx 2$) occupies a unique computational niche defined by a "High-Entropy, Low-Frequency" signature (Fig. 4, see Appendix A.2). This signature is scale-invariant (Fig. 4B), ensuring that the network receives information-rich topological structures within a slow-varying envelope. By physically restricting input trajectories to a low spectral centroid, transition dynamics function as an intrinsic spectral shaper that aligns input complexity with the spectral bias of neural networks[14]. This mechanism

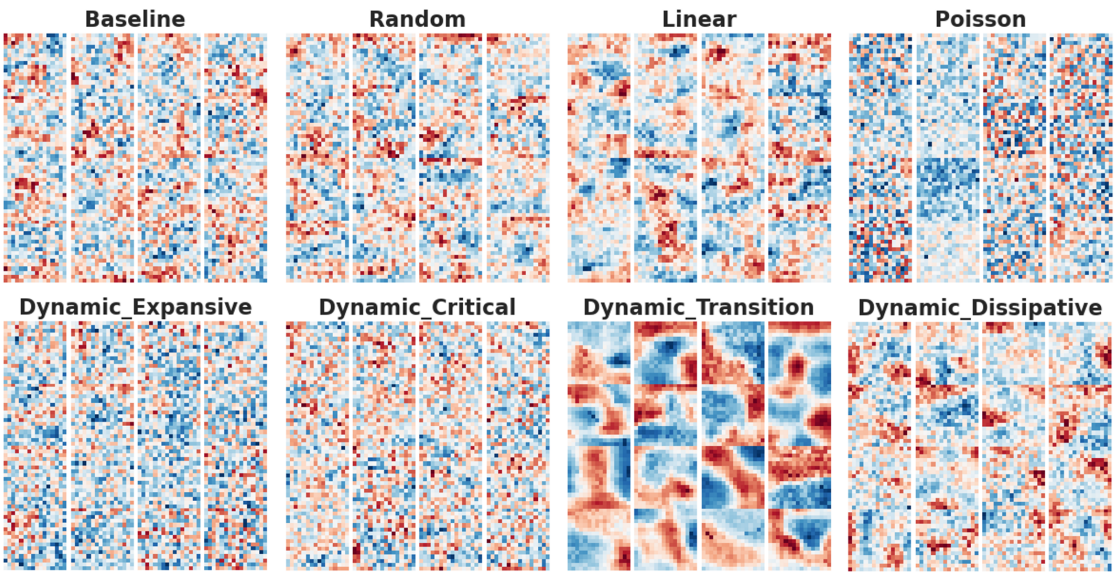

Figure 2: **Spontaneous emergence of structured receptive fields.** Visualization of learned features (receptive fields) at the network bottleneck. Red and blue pixels represent positive (excitatory) and negative (inhibitory) weights, respectively. Note that clear spatial antagonism emerges uniquely under *Transition dynamics*, whereas other regimes yield unstructured noise.

Table 1: Quantifying receptive field organization at the network bottleneck.

Encoder	Dynamics (δ)	Structure (RF Std)	Reconstruction Loss	Sparsity Loss
Dynamic Transition	**2.0**	**0.2058**	**0.003722**	**0.000868**
Dynamic Dissipative	10.0	0.0502	0.003858	0.000804
Linear	-	0.0541	0.003506	0.002270
Poisson	-	0.0492	0.003908	**0.000008**
Dynamic Expansive	-1.5	0.0435	**0.001276**	1.716117
Baseline	-	0.0417	0.003534	0.002529
Dynamic Critical	0.0	0.0367	0.003872	0.001046
Random	-	0.0341	0.003656	0.001896

is robust: a high-resolution parameter sweep reveals that this structural emergence forms a robust ridge that persists (see Appendix A.2). This confirms that feature organization arises intrinsically from the transition dynamics, acting as a physical constraint that proves more effective than explicit regularization.

Functionally, the observed structure (Figure 2) mirrors the spatial antagonism found in biological vision[9, 31] and reflects a correlation-driven consolidation process: By suppressing high-frequency noise while preserving low-frequency structure, transition dynamics amplify spatiotemporal correlations. This selective amplification biases the optimization landscape toward grouping co-varying features, yielding outcomes that are much like Hebbian association[32, 26] yet driven primarily by the temporal statistics of the input rather than explicit plasticity rules. Conversely, high-frequency chaos in the expansive regime decorrelates inputs and prevents such consolidation over time, while static or other baselines lack the necessary temporal continuity to induce structural binding. This suggests that biological dissipative constraints serve a computational role beyond energy efficiency: By naturally filtering temporal correlations, they provide a mechanistic basis for structured connectivity[4].

Furthermore, this structural emergence corroborates recent work on diffusion dynamics, which suggests a critical gap between memorization and generative abstraction[33]. Our transition dynamics position the learning process on the generative side of this divide: While the expansive regime with high-frequency chaos allows the network to memorize transient noise for rapid error reduction (the "memorization phase"), the transition regime physically imposes a temporal information bottleneck (See Table 4 and Fig. 4 in Appendix) and implements Slow Feature Analysis[34] through dynamics alone, which makes the system discard transient variations and rely on the underlying generative invariants. This re-interprets the "performance valley" observed in prior work[11] not as a failure, but as the necessary cost of abstraction.

These lines of evidence unify our findings: structured dissipative dynamics function as an active ordering principle to enable both *invariance* (Exp 1) and *organization* (Exp 2). These are not distinct phenomena but different manifestations of constraint-driven induction, which naturally leads to the third test: if low-level invariance and structural emergence both stem from temporal constraint, does this translate to high-level *behavioral robustness*?

2.4 Experiment 3: Zero-Shot Transfer to Unseen Physical Regimes

The preceding experiments established that dissipative constraints embedded in input encoding (modulated by the Duffing parameter δ) induce generalization at both representational (Exp 1) and structural (Exp 2) levels. A natural question arises: Is this principle encoding-specific, or can equivalent constraints be realized through architectural parameters? To address this, we designed a two-pronged validation targeting "unseen physical regimes" in reinforcement learning (RL) tasks [35, 36]. Unlike standard approaches relying on extrinsic domain randomization [37, 38], we posit that optimal dynamical constraints confer intrinsic robustness regardless of implementation level.

- **Encoding-Level.** We first verified that encoding-level constraints translate to behavioral robustness. Using `CartPole` [39] within a zero-shot paradigm, agents were trained on a standard "Easy" environment and evaluated on progressively difficult physical parameters (Appendix A.3). Duffing-encoded states were processed by either SNNs or MLPs to isolate the role of temporal integration.
- **Architecture-Level.** While Duffing encoding serves as an effective *probe* for investigating how dynamical constraints influence learning, it represents an artificial transformation rather than a naturally occurring mechanism. To demonstrate that the constraint principle generalizes beyond this specific implementation, we directly varied the SNN's membrane leak parameter β, which governs intrinsic temporal integration. This parallels our encoding-level analysis: just as δ controls phase space contraction in input trajectories, β controls information dissipation within the network's hidden states. We conducted systematic β-sweeps across `CartPole` (4D state, REINFORCE) and `LunarLander` [39] (8D state, PPO [40]), comparing Leaky SNNs against MLP and LSTM baselines on raw state histories without external encoding.

Both validation approaches reveal consistent patterns (Tables 2–3): At the encoding level, *SNN-Transition* ($\delta = 2.0$) achieved the lowest generalization gap (38.6 under fixed budget; 42.4 with sufficient training), significantly outperforming Expansive (87.6), Dissipative (53.1), and MLP Baseline (112.9). At the architecture level, a consistent optimum emerges: on `CartPole`, intermediate dissipation ($\beta = 0.5$) achieved the lowest gap (13.4), a 90% improvement over MLP (138.7); on the more complex `LunarLander`, the optimal regime persisted at $\beta = 0.5$ (Gap: 96.5), outperforming MLP (180.3) by 46% and LSTM (193.1) by 50%. Furthermore, we observed a non-monotonic performance profile in SNNs across the parameter spectrum: over-constrained regimes ($\beta \leq 0.3$) exhibited training instability, while unconstrained SNNs ($\beta = 1.0$) showed degraded generalization in complex tasks despite successful training. This cross-implementation convergence underscores that the efficacy of dynamical constraints is realization-agnostic: the optimal regime emerges from a characteristic timescale that optimally balances signal preservation against noise suppression.

Table 2: Encoding-level generalization comparison (`CartPole`). Duffing encoding transforms states into temporal trajectories with varying constraint strength (δ). *SNN-Transition* ($\delta = 2.0$) achieves the lowest gap, demonstrating that SNN temporal integration is required to decode constraint-induced invariants. Bold: best generalization (lowest gap).

Experiment	**Agent Group**	**Easy** (Mean ± Std)	**V. Hard** (Mean ± Std)	**Avg. Gap**	**Convergence** (Median Eps.)
Fixed Budget (2000 Eps.)	*MLP Baseline*	189.8 ± 24.8	37.3 ± 39.0	120.6	—
	MLP Exp. ($\delta = -1.5$)	179.5 ± 31.8	62.4 ± 46.1	89.3	—
	MLP Tran. ($\delta = 2.0$)	192.8 ± 11.4	74.5 ± 41.1	79.8	—
	MLP Diss. ($\delta = 10.0$)	178.6 ± 44.0	78.0 ± 35.9	56.5	—
	SNN Exp. ($\delta = -1.5$)	183.9 ± 14.5	67.0 ± 34.5	86.8	—
	SNN Tran. ($\delta = 2.0$)	169.4 ± 32.1	$\mathbf{106.1 \pm 29.6}$	**38.6**	—
	SNN Diss. ($\delta = 10.0$)	165.0 ± 29.5	87.1 ± 47.3	56.1	—
Sufficient Training	*MLP Baseline*	199.2 ± 1.5	34.6 ± 23.3	112.9	510
	MLP Exp. ($\delta = -1.5$)	198.7 ± 1.9	67.0 ± 52.2	105.4	970
	MLP Tran. ($\delta = 2.0$)	198.0 ± 2.2	80.6 ± 52.6	72.2	1360
	MLP Diss. ($\delta = 10.0$)	198.9 ± 1.1	67.2 ± 41.3	81.9	840
	SNN Exp. ($\delta = -1.5$)	198.3 ± 1.4	70.3 ± 48.3	87.6	2870
	SNN Tran. ($\delta = 2.0$)	194.7 ± 6.4	$\mathbf{126.3 \pm 34.2}$	**42.4**	2060
	SNN Diss. ($\delta = 10.0$)	193.6 ± 8.2	107.6 ± 32.8	53.1	2270

Table 3: Architecture-level generalization comparison. Varying membrane leak β on raw state inputs (no external encoding) reveals a non-monotonic optimum: intermediate dissipation ($\beta \approx 0.5$) yields best generalization. Note that over-constrained regimes ($\beta \leq 0.3$) fail to converge reliably. Bold: best generalization (lowest gap).

Experiment	Agent Group	Easy (Mean ± Std)	V. Hard (Mean ± Std)	Avg. Gap	Convergence (Median Eps.)
CartPole	*MLP (baseline)*	199.0 ± 2.0	60.3 ± 50.4	138.7	100
	LSTM (baseline)	169.0 ± 60.5	127.0 ± 65.2	42.0	50
	Leaky SNN ($\beta = 0.1$)	$8.4 \pm 0.1^{\dagger}$	15.4 ± 0.3	—	1050
	Leaky SNN ($\beta = 0.3$)	$10.9 \pm 4.9^{\dagger}$	20.0 ± 8.9	—	1050
	Leaky SNN ($\beta = 0.5$)	195.5 ± 2.7	182.1 ± 3.7	**13.4**	2500
	Leaky SNN ($\beta = 0.7$)	198.2 ± 1.5	179.0 ± 10.7	19.2	1150
	Leaky SNN ($\beta = 0.9$)	196.9 ± 2.0	176.7 ± 8.5	20.2	950
	Leaky SNN ($\beta = 1.0$)	199.9 ± 0.1	186.4 ± 6.4	13.5	850
LunarLander	*MLP (baseline)*	238.9 ± 11.2	58.6 ± 40.5	180.3	400
	LSTM (baseline)	220.0 ± 14.7	26.9 ± 27.8	193.1	200
	Leaky SNN ($\beta = 0.1$)	$-79.3 \pm 289.3^{\dagger}$	-209.8 ± 199.9	—	3600
	Leaky SNN ($\beta = 0.3$)	$153.6 \pm 123.8^{*}$	28.5 ± 24.2	125.1	3600
	Leaky SNN ($\beta = 0.5$)	217.9 ± 13.9	121.4 ± 18.0	**96.5**	2400
	Leaky SNN ($\beta = 0.7$)	234.8 ± 13.4	122.4 ± 19.8	112.4	2000
	Leaky SNN ($\beta = 0.9$)	224.8 ± 20.6	96.4 ± 38.9	128.4	2000
	Leaky SNN ($\beta = 1.0$)	229.1 ± 10.7	125.0 ± 21.9	104.1	1600

†Training failed to converge. *High variance, partial convergence only.

Furthermore, the non-monotonic performance profile observed in Table 3 helps disentangle a critical confound: does robustness originate from the temporal constraint itself, or merely from the "memory" capacity inherent in SNN membrane dynamics[13]? If memory alone drove generalization, performance should monotonically increase with information retention (high β). Instead, the observed degradation (or failure) at $\beta \approx 1.0$ indicates that intermediate dissipation implements a necessary temporal information bottleneck, filtering transient noise while preserving invariant structure. This interpretation is further reinforced by evidence from recurrent variants (*RLeaky SNNs*, see Appendix A.3.2): despite possessing additional memory pathways, these networks did not exhibit improved generalization. On the contrary, they displayed a significantly narrower stability window, failing to converge in unconstrained regimes ($\beta \approx 1.0$) where feedforward Leaky SNNs still functioned. This fragility suggests that excess memory without sufficient constraint destabilizes learning, revealing an inherent duality of the dynamical parameter, as it simultaneously governs both information retention and phase space contraction.

In conclusion, these multi-level validations provide converging evidence for our central thesis: *temporal constraint breeds generalization*. Structured dissipation functions not as a limitation but as a temporal inductive bias, compelling networks to extract robust, invariant representations regardless of whether constraints are imposed externally via (specific) encoding or internally via architecture.

3 Discussion

Related Work Recent literature increasingly frames physical constraints not as computational limitations, but as essential inductive biases for intelligence. These constraints can be implemented at different system levels: through architectural design[41], learnable internal dynamics[42], or temporal hierarchies[43, 44]. Our work reveals a complementary mechanism: temporal dissipation embedded in input dynamics functions as an inductive bias, operating independently of architectural specification or learnable parameters. This finding aligns with Slow Feature Analysis (SFA)[34], which posits that invariance emerges from slowly varying signals. However, unlike SFA which explicitly optimizes for this objective, we demonstrate that dissipative dynamics naturally impose spectral compression as a physical implementation of temporal invariance principles. Together, these approaches suggest that constraints can operate across multiple system levels (input, architecture, dynamics), and our results indicate that temporal constraints alone can induce the invariant representations required for out-of-distribution generalization[45, 15].

From Generalization to Specialization This framework provides a theoretical justification for the "performance valley" discussed earlier[11]. What appears to be a functional trade-off is mathematically substantiated by our complementary PAC-Bayes analysis[46] (Appendix A.4): By bounding the generalization error as a function of the posterior's alignment with the prior, this analysis reveals that the transition regime minimizes the trade-off between empirical fit (training error) and structural stability (Kullback–Leibler divergence). This theoretical insight re-interprets the observed performance dip as a necessary cost of abstraction, i.e., the increased training effort. As evidenced by the delayed convergence in our behavioral tasks (Exp 3, Appendix Figure 6), extracting robust invariants is inherently more computationally

demanding than the rapid memorization of transient patterns. In other words, the very regime that appears suboptimal from a purely efficiency-centric view is proven to be globally optimal for learning stable, transferable solutions.

This finding not only aligns with the "Edge of Chaos" theory of computational capacity [47, 48] but also extends its theoretical scope by explicitly identifying state temporal constraints as an active regularization mechanism. Taken together, we can impose a broader picture: rather than a static optimum for all computation, the neural computation landscape should be viewed as a continuous dynamical spectrum. In this view, the transition regime acts as an adaptability reservoir, maintaining the scale-invariant complexity required to distill robust invariants. Moving away from this center, adjacent regimes serve distinct downstream objectives: shifting towards expansive dynamics ($\Sigma\lambda_i > 0$) maximizes discriminability for high-stakes inference, or collapsing towards dissipative dynamics ($\Sigma\lambda_i \ll 0$) ensures stability and energy minimization.

Therefore, this implies that the neural system should be formalized as a non-autonomous dynamical system, where the "optimal" configuration is not a fixed point (or fixed state) but a dynamical spectrum through the phase space. From this perspective, the neuron's dynamical properties are not intrinsic constants but mutable state variables that evolve in response to environmental demands and internal energy budgets. For machine learning, this necessitates a paradigm shift from architectural search to system dynamic regime navigation: robust intelligence emerges from systems that possess the agency to modulate their internal states dynamically. Time thus acts as a flexible resource that acts as the fundamental balance between retaining specific data fidelity and distilling generalizable understanding. The transition regime plays the role of a generalization optimum, which emerges not from maximizing memory, but from aligning the integration timescale to task-relevant invariants. This reconciliation offers a potential resolution to the apparent paradox that biological neural systems, despite vast memory capacity or energy efficiency, exhibit robust generalization: sparse coding and metabolic constraints may represent evolutionary solutions that naturally situate neural dynamics within the transition regime.

Limitations and Future Directions While this work demonstrates the efficacy of dynamical constraints, our implementation primarily employs a fixed oscillator system as a probe for mechanistic discovery: By isolating specific dynamical regimes, we establish a controlled environment for qualitative analysis, yet this represents only a static instantiation; the performance reported herein likely constitutes a lower bound on the paradigm's potential. Although our architecture-level experiments (β-sweeps) demonstrate that the constraint principle generalizes beyond Duffing encoding, both implementations employ only fixed parameters. Meanwhile, biological systems exhibit heterogeneous time constants across neural populations; exploring learned or adaptive β distributions represents a natural extension. The critical observation is that the transition regime is defined not by a specific parameter value, but by a spectral signature: the combination of high entropy (preserved information complexity) and low dominant frequency (alignment with neural networks' spectral bias). As demonstrated in our scale-space analysis (Fig.4B), this signature remains stable across timescales ($T_{\max}$) and sampling densities (N), suggesting an intrinsic property of the dynamical regime rather than a parameter-tuning artifact. Nevertheless, we acknowledge that verifying this regime-generality across alternative dynamical systems remains an important direction for future work. Ultimately, this study suggests that the path toward robust AI lies not just in scaling model parameters, but in mastering the temporal constraints that shape the learning landscape itself.

4 Conclusion

This work identifies dynamical constraints as a temporal inductive bias. By optimally filtering transient noise while retaining topological complexity, this regime fosters robust invariance. We show that this capability is architecturally distinct: it requires SNN-based temporal integration to decode constraints that static baselines perceive as noise. Consequently, we propose that biological efficiency constraints are not computational hurdles but evolutionary pressures that drive the emergence of robust, generalizable intelligence.

Acknowledgments

This work was supported by the Georg Nemetschek Institute at the Technical University of Munich through the TUM GNI Postdoc Program.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[3] Sara Hooker. "The hardware lottery". In: *Communications of the ACM* 64.12 (2021), pp. 58–65.

[4] György Buzsáki. *Rhythms of the Brain*. Oxford university press, 2006.

[5] Sohan Shankar et al. "Bridging brains and machines: A unified frontier in neuroscience, artificial intelligence, and neuromorphic systems". In: *arXiv preprint arXiv:2507.10722* (2025).

[6] Jared Kaplan et al. "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361* (2020).

[7] Behnam Neyshabur et al. "Exploring generalization in deep learning". In: *Advances in neural information processing systems* 30 (2017).

[8] David Attwell and Simon B Laughlin. "An energy budget for signaling in the grey matter of the brain". In: *Journal of Cerebral Blood Flow & Metabolism* 21.10 (2001), pp. 1133–1145.

[9] Bruno A Olshausen and David J Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583 (1996), pp. 607–609.

[10] Rufin Van Rullen and Simon J Thorpe. "Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex". In: *Neural computation* 13.6 (2001), pp. 1255–1283.

[11] Xia Chen. "Dynamical Alignment: A Principle for Adaptive Neural Computation". In: *arXiv preprint arXiv:2508.10064* (2025).

[12] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. "Towards spike-based machine intelligence with neuromorphic computing". In: *Nature* 575.7784 (2019), pp. 607–617.

[13] Jason K Eshraghian et al. "Training spiking neural networks using lessons from deep learning". In: *Proceedings of the IEEE* 111.9 (2023), pp. 1016–1054.

[14] Nasim Rahaman et al. "On the spectral bias of neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 5301–5310.

[15] Anirudh Goyal and Yoshua Bengio. "Inductive biases for deep learning of higher-level cognition". In: *Proceedings of the Royal Society A* 478.2266 (2022), p. 20210068.

[16] Alex Hernández-García and Peter König. "Data augmentation instead of explicit regularization". In: *arXiv preprint arXiv:1806.03852* (2018).

[17] Randall Balestriero, Leon Bottou, and Yann LeCun. "The effects of regularization and data augmentation are class dependent". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 37878–37891.

[18] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[19] Alan Wolf et al. "Determining Lyapunov exponents from a time series". In: *Physica D: nonlinear phenomena* 16.3 (1985), pp. 285–317.

[20] Ivana Kovacic and Michael J Brennan. *The Duffing equation: nonlinear oscillators and their behaviour*. John Wiley & Sons, 2011.

[21] E. Alpaydin and C. Kaynak. *Optical Recognition of Handwritten Digits*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50P49. 1998.

[22] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[23] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. "How does the brain solve visual object recognition?" In: *Neuron* 73.3 (2012), pp. 415–434.

[24] Shailaja Akella et al. "Deciphering neuronal variability across states reveals dynamic sensory encoding". In: *Nature Communications* 16.1 (2025), p. 1768.

[25] Daniel LK Yamins et al. "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the national academy of sciences* 111.23 (2014), pp. 8619–8624.

[26] Horace B Barlow et al. "Possible principles underlying the transformation of sensory messages". In: *Sensory communication* 1.01 (1961), pp. 217–233.

[27] Daniel Auge et al. "A survey of encoding techniques for signal processing in spiking neural networks". In: *Neural Processing Letters* 53.6 (2021), pp. 4693–4710.

[28] Wenzhe Guo et al. "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems". In: *Frontiers in Neuroscience* 15 (2021), p. 638474.

[29] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[30] Adam Coates, Andrew Ng, and Honglak Lee. "An analysis of single-layer networks in unsupervised feature learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 215–223.

[31] David Ferster and Kenneth D Miller. "Neural mechanisms of orientation selectivity in the visual cortex". In: *Annual review of neuroscience* 23.1 (2000), pp. 441–471.

[32] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[33] Tony Bonnaire et al. "Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training". In: *arXiv preprint arXiv:2505.17638* (2025).

[34] Laurenz Wiskott and Terrence J Sejnowski. "Slow feature analysis: Unsupervised learning of invariances". In: *Neural computation* 14.4 (2002), pp. 715–770.

[35] Charles Packer et al. "Assessing generalization in deep reinforcement learning". In: *arXiv preprint arXiv:1810.12282* (2018).

[36] Robert Kirk et al. "A survey of zero-shot generalisation in deep reinforcement learning". In: *Journal of Artificial Intelligence Research* 76 (2023), pp. 201–264.

[37] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30.

[38] Xue Bin Peng et al. "Sim-to-real transfer of robotic control with dynamics randomization". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 3803–3810.

[39] Mark Towers et al. "Gymnasium: A standard interface for reinforcement learning environments". In: *arXiv preprint arXiv:2407.17032* (2024).

[40] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[41] Quercus Hernández et al. "Thermodynamics-informed graph neural networks". In: *IEEE Transactions on Artificial Intelligence* 5.3 (2022), pp. 967–976.

[42] Shao-Qun Zhang et al. "On the intrinsic structures of spiking neural networks". In: *Journal of Machine Learning Research* 25.194 (2024), pp. 1–74.

[43] Xiang Cheng et al. "Finite meta-dynamic neurons in spiking neural networks for spatio-temporal learning". In: *arXiv preprint arXiv:2010.03140* (2020).

[44] Filippo Moro et al. "The role of temporal hierarchy in spiking neural networks". In: *arXiv preprint arXiv:2407.18838* (2024).

[45] Weijian Deng, Stephen Gould, and Liang Zheng. "On the strong correlation between model invariance and generalization". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 28052–28067.

[46] David A McAllester. "Some pac-bayesian theorems". In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 230–234.

[47] Chris G Langton. "Computation at the edge of chaos: Phase transitions and emergent computation". In: *Physica D: nonlinear phenomena* 42.1-3 (1990), pp. 12–37.

[48] Nils Bertschinger, Thomas Natschläger, and Robert Legenstein. "At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks". In: *Advances in neural information processing systems* 17 (2004).

A Appendix

Justification of Dynamical System & Metric Selection

The selection of the modified Duffing oscillator in this study is grounded in prior investigation of dynamical alignment[1]. A comparative analysis across six classical chaotic attractors (Lorenz, Rössler, Chua, etc.) established that the computational regime of a neural network is primarily governed by the global evolution of phase space volume rather than specific geometric manifolds. To isolate this critical factor from the geometric confounds inherent in classical attractors, we adopted a coupled Duffing system as a standardized probe. Each input feature x_i initializes a three-dimensional dynamical system. The initial state (x_0, y_0, z_0) is set as:

$$\begin{aligned} x_0 &= x_i \\ y_0 &= 0.2 \cdot x_i \\ z_0 &= -x_i \end{aligned} \tag{1}$$

This linear mapping ensures that different feature values generate distinct trajectories, preserving discriminability while providing a consistent initialization scheme across the feature space.

The initial conditions evolve according to the coupled oscillator dynamics:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\alpha x - \beta x^3 - \delta y + \gamma z \\ \dot{z} &= -\omega x - \delta z + \gamma x y \end{aligned}$$

where hyperparameters are fixed to $\alpha = 2.0$, $\beta = 0.1$, $\gamma = 0.1$, $\omega = 1.0$. The system evolves for a total time T and is sampled at N discrete timesteps, yielding a trajectory $\phi_\delta(x_i, t) \in \mathbb{R}^{N\times 3}$. For a d-dimensional input, the complete encoded representation is $\Phi_\delta(\mathbf{x}, t) \in \mathbb{R}^{d\times N\times 3}$. This parametrically controllable system allows for the continuous tuning of global phase space dynamics via a single parameter δ, effectively decoupling the rate of phase space contraction from the topological structure of the attractor.

To characterize the physical behavior of this system, we systematically mapped its dynamical properties across the control spectrum. Table 4 presents the relationship between δ and three key dynamical indicators: the Maximum Lyapunov Exponent ($\lambda_{\max}$), the Lyapunov Sum ($\Sigma\lambda_i$)[2], and Active Information Storage (AIS)[3]. As evidenced by the data, the parameter δ provides monotonic control over the phase space contraction rate ($\Sigma\lambda_i$), spanning from the expansive regime ($\Sigma\lambda_i > 0$) to the strongly dissipative regime ($\Sigma\lambda_i \ll 0$).

Table 4: Dynamical property reference of the mixed oscillator system[1]. The parameter δ regulates global phase space dynamics. AIS (Active Information Storage) values in *italics* highlight the information bottleneck in the transition regime ($\delta \approx 2.0$).

δ	**-1.5**	**-1.0**	**-0.6**	**-0.3**	**-0.15**	**0**	**0.15**	**0.3**	**0.6**	**1.0**	**1.5**	**2.0**	**2.5**	**4.0**	**5.0**	**7.0**	**10.0**
$\lambda_{\max}$	1.17	0.79	0.44	0.24	0.16	0.08	0	-0.07	-0.22	-0.41	-0.70	-0.88	-1.15	-0.55	-0.42	-0.29	-0.20
$\Sigma\lambda_i$	3	2	1.2	0.6	0.3	0	-0.3	-0.6	-1.2	-2	-3	-4	-5	-8	-10	-14	-20
AIS	2.99	2.95	3.05	2.80	3.02	2.97	2.99	2.99	2.66	2.15	*1.62*	*1.06*	*1.09*	2.07	2.44	2.84	3.08

These empirical metrics justify our definition of the *Transition* regime ($\delta \approx 2.0$) as a unique computational state. While the Lyapunov sum decreases monotonically with δ, the AIS exhibits a distinct non-monotonic behavior, reaching a global minimum in the region $\delta \in [1.5, 2.5]$. Specifically, at $\delta = 2.0$, the system enters a state characterized simultaneously by weak dissipation ($\Sigma\lambda_i = -4$), indicating phase space contraction, and a critical information bottleneck. In this state, the AIS drops to its lowest level (≈ 1.06), which is significantly lower than both the expansive (≈ 2.99) and strongly dissipative (≈ 3.08) extremes. This sharp dip in information storage capacity corresponds to a state of minimal short-term memory retention. This characteristic is consistent with a filtering mechanism that suppresses transient noise, corroborating the emergence of robust, invariant features described in the main text.

For our experiments, we span two complementary configurations to demonstrate the robustness of the principle:

- *High-Resolution Mapping (Exp 1):* To rigorously map the generalization landscape, we employed a dense sampling regime ($T = 4.0, N = 30$). This setting minimizes numerical drift to ensure precise topological mapping.
- *Efficient Application (Exp 2 & 3):* For downstream tasks (Receptive Field Learning and RL), we adopted a computationally efficient sparse regime ($T = 8.0, N = 5$). This configuration accommodates the stricter computational budget inherent in these iterative training loops while retaining sufficient dynamical complexity.

As evidenced by our scale-space analysis (Fig. 4B), the mechanism is regime-dependent rather than parameter-specific; the critical "low-frequency, high-entropy" signature of the transition regime persists across varying observation scales (N) and evolution times (T_{max}), the mechanism governed by δ remains invariant: tuning δ continuously traverses the spectrum from expansive ($\delta < 0$) to dissipative ($\delta \gg 0$) dynamics, with the transition regime ($\delta \approx 2.0$) consistently emerging as the critical state for generalization.

A.1 Experiment 1: Cross-Encoding Generalization Details

Dataset and Setup. We use the `sklearn.datasets.load_digits` dataset (64 features, 10 classes)[4, 5], split into 70% training, 15% validation, and 15% testing. Input features $\mathbf{x}$ are first normalized using StandardScaler. Networks are trained on data encoded with one δ_{train} value, then tested on 12 different encodings: $\delta_{\text{test}} \in \{-1.5, -1.0, -0.3, 0.0, 0.3, 1.0, 1.5, 2.0, 2.5, 5.0, 7.0, 10.0\}$. The detailed quantitative generalization matrices (Mean $\pm$ Std %) for all architectures are provided in Tables 5 - 9.

Training. LIF neurons use learnable thresholds and membrane decay $\beta = 0.95$. Training uses surrogate gradients (`surrogate.fast_sigmoid` with `slope=25`)[6] with the Adam optimizer (lr=1e-4)[7] for 200 epochs, with early stopping (patience=10) based on validation accuracy.

Overall Correlation Across All Train-Test Pairs. Table 10 presents the Pearson correlation coefficients between layer-wise CV and OOD accuracy across all 1,440 samples (12 train encodings $\times$ 12 test encodings $\times$ 10 runs). All layers exhibit significant negative correlations, with correlation strength decreasing hierarchically from Layer 1 to Layer 3. This pattern suggests that early-layer stability is most predictive of generalization performance. Statistical significance was assessed using two-tailed tests with $\alpha = 0.05$.

Table 5: Cross-encoding generalization (mean ± std %) for the SNN model. Rows indicate δ_{train}, columns indicate δ_{test}. Diagonal (in-distribution) results are bolded.

Train δ	**Test** δ											
	-1.5	**-1.0**	**-0.3**	**0.0**	**0.3**	**1.0**	**1.5**	**2.0**	**2.5**	**5.0**	**7.0**	**10.0**
-1.5	**92.9 ± 1.9**	92.0 ± 2.0	56.1 ± 10.7	38.7 ± 11.6	29.0 ± 9.6	21.2 ± 6.9	18.6 ± 6.7	17.9 ± 6.5	18.0 ± 6.6	17.6 ± 7.2	17.9 ± 7.2	17.6 ± 6.6
-1.0	90.9 ± 5.9	**91.7 ± 6.3**	71.4 ± 5.8	51.0 ± 7.9	36.1 ± 8.4	21.1 ± 6.8	17.3 ± 6.3	16.1 ± 6.0	15.1 ± 5.6	14.4 ± 5.0	14.1 ± 5.0	14.1 ± 4.5
-0.3	85.8 ± 2.8	90.3 ± 2.2	**95.2 ± 1.0**	94.5 ± 1.0	92.4 ± 1.9	87.8 ± 4.1	85.3 ± 5.6	84.4 ± 6.4	82.3 ± 6.7	79.3 ± 7.1	77.9 ± 8.6	76.8 ± 8.2
0.0	81.4 ± 6.1	86.6 ± 3.6	94.3 ± 1.8	**94.6 ± 1.8**	94.9 ± 1.5	92.8 ± 1.9	92.3 ± 2.3	91.2 ± 2.6	90.9 ± 2.2	89.2 ± 2.6	89.0 ± 2.6	88.5 ± 2.3
0.3	81.0 ± 2.6	86.2 ± 1.8	93.5 ± 1.6	94.8 ± 1.7	**94.7 ± 1.8**	94.2 ± 1.7	93.4 ± 2.3	93.5 ± 2.8	93.1 ± 2.6	92.2 ± 2.1	91.9 ± 2.5	90.9 ± 2.4
1.0	79.6 ± 2.8	84.7 ± 2.4	92.6 ± 1.8	94.4 ± 1.5	94.3 ± 1.7	**94.5 ± 1.9**	94.2 ± 1.8	93.9 ± 2.2	93.6 ± 2.2	93.3 ± 2.3	92.9 ± 2.8	92.7 ± 2.4
1.5	72.5 ± 4.7	78.6 ± 4.4	90.3 ± 1.9	92.9 ± 1.7	94.4 ± 1.8	94.8 ± 1.7	**94.6 ± 2.0**	94.5 ± 2.1	94.4 ± 1.8	93.5 ± 2.1	93.2 ± 2.0	93.0 ± 2.1
2.0	70.9 ± 8.1	76.9 ± 8.5	90.3 ± 2.6	92.8 ± 1.7	93.8 ± 1.9	94.3 ± 1.9	94.3 ± 1.8	**94.2 ± 2.0**	93.9 ± 2.4	93.6 ± 2.3	93.4 ± 2.0	93.3 ± 2.0
2.5	68.5 ± 6.9	74.4 ± 6.1	90.0 ± 2.2	93.2 ± 1.8	93.7 ± 1.4	94.4 ± 1.4	94.8 ± 1.2	94.6 ± 1.1	**94.8 ± 1.1**	94.6 ± 1.4	94.4 ± 1.5	93.9 ± 1.4
5.0	65.1 ± 6.9	71.1 ± 5.8	86.3 ± 3.2	89.9 ± 2.3	91.5 ± 2.3	93.4 ± 1.7	93.8 ± 1.6	94.3 ± 1.8	94.1 ± 1.8	**94.7 ± 1.3**	94.7 ± 1.8	94.2 ± 1.7
7.0	66.9 ± 8.9	71.7 ± 7.6	86.0 ± 4.8	89.9 ± 2.9	91.5 ± 2.1	93.3 ± 1.4	93.6 ± 1.2	93.7 ± 1.9	94.2 ± 1.6	94.5 ± 1.6	**94.5 ± 1.4**	94.6 ± 1.5
10.0	69.1 ± 6.3	74.2 ± 5.5	85.7 ± 3.0	89.6 ± 2.4	91.2 ± 2.1	92.5 ± 2.3	92.9 ± 2.4	93.5 ± 2.2	93.5 ± 1.9	94.1 ± 1.7	94.5 ± 1.9	**94.6 ± 2.1**

Table 6: Cross-encoding generalization (mean ± std %) for the MLP (Avg-Pool) model. Rows indicate δ_{train}, columns indicate δ_{test}. Diagonal (in-distribution) results are bolded.

Train δ	**Test** δ											
	-1.5	**-1.0**	**-0.3**	**0.0**	**0.3**	**1.0**	**1.5**	**2.0**	**2.5**	**5.0**	**7.0**	**10.0**
-1.5	**94.9 ± 1.3**	94.8 ± 1.4	67.1 ± 9.4	39.2 ± 13.4	23.7 ± 9.7	15.8 ± 5.1	12.0 ± 3.8	10.4 ± 3.0	8.9 ± 2.9	6.3 ± 4.0	5.6 ± 4.1	4.9 ± 4.1
-1.0	95.0 ± 1.8	**95.0 ± 1.7**	93.2 ± 1.6	87.9 ± 2.5	69.3 ± 6.9	16.5 ± 4.0	5.3 ± 2.0	2.0 ± 1.2	1.4 ± 1.2	0.6 ± 0.7	0.6 ± 0.8	0.5 ± 0.6
-0.3	93.8 ± 1.9	94.1 ± 1.9	**94.5 ± 1.9**	93.9 ± 2.0	92.7 ± 2.2	19.7 ± 9.0	2.5 ± 3.2	0.9 ± 1.5	0.5 ± 0.9	0.2 ± 0.3	0.2 ± 0.3	0.2 ± 0.3
0.0	90.6 ± 8.0	90.7 ± 7.9	90.8 ± 8.5	**91.1 ± 8.2**	91.1 ± 7.6	90.6 ± 7.1	88.3 ± 6.5	83.8 ± 5.9	64.1 ± 7.5	0.5 ± 0.9	0.2 ± 0.2	0.1 ± 0.2
0.3	91.9 ± 2.5	92.1 ± 2.6	92.5 ± 3.0	92.8 ± 2.9	**92.7 ± 2.7**	92.0 ± 2.3	91.2 ± 1.9	88.6 ± 1.9	74.7 ± 10.6	0.4 ± 0.6	0.2 ± 0.4	0.1 ± 0.3
1.0	90.7 ± 3.3	90.8 ± 3.3	91.0 ± 3.3	91.3 ± 3.4	91.5 ± 3.6	**91.8 ± 3.3**	91.7 ± 3.4	91.7 ± 3.4	91.6 ± 3.5	91.7 ± 3.6	91.7 ± 3.5	91.7 ± 3.5
1.5	91.0 ± 2.4	91.0 ± 2.5	91.3 ± 2.6	92.1 ± 2.6	92.1 ± 2.9	92.1 ± 2.8	**92.0 ± 2.9**	91.9 ± 2.9	91.9 ± 2.9	91.8 ± 3.0	91.8 ± 2.9	91.9 ± 2.9
2.0	86.5 ± 14.5	86.6 ± 14.8	86.9 ± 14.1	87.0 ± 13.9	87.2 ± 14.0	87.6 ± 13.8	87.7 ± 13.9	**87.6 ± 14.0**	87.6 ± 14.0	87.6 ± 14.2	87.4 ± 14.3	87.3 ± 14.2
2.5	88.5 ± 9.9	88.6 ± 9.7	88.9 ± 9.6	89.2 ± 9.4	89.5 ± 9.7	89.4 ± 9.5	89.3 ± 9.4	89.4 ± 9.2	**89.2 ± 9.4**	89.2 ± 9.5	89.1 ± 9.3	89.0 ± 9.3
5.0	91.3 ± 2.7	91.3 ± 2.6	92.0 ± 2.2	92.0 ± 1.9	92.0 ± 1.8	92.0 ± 1.9	92.1 ± 2.0	92.1 ± 2.1	92.1 ± 2.1	**92.1 ± 2.2**	92.1 ± 2.1	92.1 ± 2.1
7.0	87.2 ± 11.8	87.4 ± 11.5	87.4 ± 10.8	87.7 ± 10.5	87.7 ± 10.3	88.1 ± 10.3	88.2 ± 10.3	88.3 ± 10.2	88.3 ± 10.0	88.4 ± 10.3	**88.5 ± 10.3**	88.5 ± 10.3
10.0	92.3 ± 2.2	92.3 ± 2.3	92.4 ± 2.1	92.4 ± 1.9	92.5 ± 1.9	92.9 ± 1.7	92.9 ± 1.8	92.9 ± 1.8	92.9 ± 1.7	93.2 ± 1.7	93.3 ± 1.7	**93.3 ± 1.7**

Table 7: Cross-encoding generalization (mean ± std %) for the MLP (T-last) model. Rows indicate δ_{train}, columns indicate δ_{test}. Diagonal (in-distribution) results are bolded.

Train δ	**Test** δ											
	-1.5	**-1.0**	**-0.3**	**0.0**	**0.3**	**1.0**	**1.5**	**2.0**	**2.5**	**5.0**	**7.0**	**10.0**
-1.5	**92.3 ± 2.3**	90.2 ± 2.6	21.8 ± 2.3	11.0 ± 1.5	9.4 ± 2.2	10.0 ± 2.7	10.2 ± 2.2	10.5 ± 1.6	10.5 ± 1.6	11.3 ± 1.7	11.4 ± 1.9	11.4 ± 1.9
-1.0	94.4 ± 1.9	**94.9 ± 1.8**	89.1 ± 3.3	68.4 ± 9.9	34.1 ± 12.7	13.1 ± 5.7	11.1 ± 3.2	9.4 ± 1.7	9.3 ± 1.7	13.3 ± 5.7	15.3 ± 7.1	18.8 ± 8.3
-0.3	94.1 ± 2.2	94.8 ± 1.9	**95.0 ± 2.3**	94.4 ± 2.3	93.2 ± 2.2	45.8 ± 15.1	13.1 ± 4.1	6.9 ± 2.8	7.0 ± 3.2	44.1 ± 14.3	72.5 ± 10.2	85.2 ± 5.5
0.0	93.4 ± 1.6	94.0 ± 1.5	94.1 ± 1.8	**94.2 ± 1.8**	93.9 ± 2.0	62.3 ± 12.7	14.2 ± 6.8	7.4 ± 4.2	8.2 ± 3.8	54.4 ± 12.8	82.9 ± 4.8	89.8 ± 2.1
0.3	0.1 ± 0.2	0.3 ± 0.4	86.3 ± 11.4	88.7 ± 10.9	**89.3 ± 10.3**	86.7 ± 9.9	50.0 ± 4.7	12.9 ± 4.1	7.1 ± 2.7	31.6 ± 5.9	75.7 ± 8.7	86.0 ± 9.8
1.0	0.0 ± 0.0	0.0 ± 0.1	80.1 ± 16.0	81.0 ± 14.8	81.3 ± 14.3	**81.5 ± 14.8**	70.6 ± 16.2	15.6 ± 6.7	6.3 ± 3.8	36.3 ± 13.4	77.4 ± 17.1	80.6 ± 16.8
1.5	0.1 ± 0.4	0.1 ± 0.4	0.1 ± 0.4	0.9 ± 0.7	71.9 ± 18.9	72.7 ± 18.9	**73.0 ± 19.5**	67.6 ± 17.7	6.8 ± 3.9	0.2 ± 0.3	0.2 ± 0.4	0.2 ± 0.4
2.0	1.3 ± 1.7	1.1 ± 1.4	1.3 ± 2.6	3.7 ± 3.4	24.7 ± 12.6	26.0 ± 11.7	24.8 ± 12.0	**22.4 ± 12.5**	7.8 ± 3.5	2.1 ± 3.9	2.0 ± 3.9	2.0 ± 3.9
2.5	25.4 ± 4.9	23.2 ± 4.9	6.3 ± 3.7	4.7 ± 4.0	4.8 ± 4.2	6.3 ± 4.7	7.5 ± 4.6	10.4 ± 1.4	**10.6 ± 1.0**	10.8 ± 1.6	8.5 ± 4.1	7.9 ± 4.7
5.0	88.1 ± 3.3	87.7 ± 3.2	87.9 ± 3.5	74.4 ± 5.0	0.2 ± 0.2	0.1 ± 0.2	0.2 ± 0.3	0.8 ± 1.0	12.1 ± 3.9	**89.6 ± 2.6**	89.4 ± 2.8	89.2 ± 2.9
7.0	87.9 ± 6.4	87.4 ± 6.3	87.5 ± 6.5	70.0 ± 10.3	0.3 ± 0.5	0.3 ± 0.5	0.4 ± 0.7	1.4 ± 2.3	10.2 ± 1.5	88.9 ± 6.0	**89.1 ± 6.4**	89.0 ± 6.3
10.0	91.0 ± 2.7	90.7 ± 2.6	91.1 ± 2.6	72.0 ± 5.0	0.4 ± 0.4	0.2 ± 0.2	0.4 ± 0.4	1.4 ± 0.9	10.8 ± 3.1	92.1 ± 2.7	92.9 ± 2.9	**92.7 ± 2.8**

Table 8: Cross-encoding generalization (mean ± std %) for the LSTM model. Rows indicate δ_{train}, columns indicate δ_{test}. Diagonal (in-distribution) results are bolded.

Train δ	**Test** δ											
	-1.5	**-1.0**	**-0.3**	**0.0**	**0.3**	**1.0**	**1.5**	**2.0**	**2.5**	**5.0**	**7.0**	**10.0**
-1.5	**87.6 ± 1.6**	87.6 ± 2.1	79.3 ± 3.9	72.2 ± 4.2	64.8 ± 4.9	51.3 ± 6.3	44.0 ± 7.6	37.3 ± 8.0	32.3 ± 7.3	20.4 ± 6.4	16.7 ± 5.7	14.1 ± 5.1
-1.0	87.4 ± 2.8	**89.6 ± 2.4**	83.0 ± 3.7	69.7 ± 7.0	54.7 ± 8.1	33.4 ± 7.1	25.8 ± 6.6	20.9 ± 5.8	18.0 ± 4.2	11.4 ± 3.0	9.2 ± 2.9	8.1 ± 2.6
-0.3	71.4 ± 3.8	82.1 ± 3.2	**94.0 ± 2.2**	89.0 ± 1.9	71.1 ± 4.3	25.6 ± 6.3	16.1 ± 6.7	12.7 ± 5.3	10.7 ± 4.8	9.7 ± 3.7	10.0 ± 3.6	10.4 ± 2.8
0.0	60.2 ± 6.5	72.9 ± 5.4	92.4 ± 2.0	**93.5 ± 1.5**	85.8 ± 2.9	32.8 ± 9.0	16.4 ± 7.4	12.3 ± 7.2	10.7 ± 6.3	9.0 ± 5.3	9.0 ± 5.5	9.0 ± 5.5
0.3	7.5 ± 3.2	12.3 ± 4.9	86.2 ± 4.2	93.3 ± 1.5	**93.4 ± 1.6**	68.8 ± 6.1	35.7 ± 7.0	19.8 ± 5.1	12.5 ± 4.1	4.0 ± 1.5	2.7 ± 1.2	2.4 ± 1.1
1.0	5.1 ± 2.4	7.0 ± 3.4	60.6 ± 5.1	78.5 ± 4.3	86.5 ± 4.0	**92.1 ± 1.6**	80.1 ± 3.3	42.3 ± 11.0	24.2 ± 9.3	7.5 ± 4.7	5.3 ± 4.3	3.9 ± 3.6
1.5	3.0 ± 1.3	4.7 ± 2.5	46.5 ± 7.9	57.1 ± 8.5	64.5 ± 6.6	87.5 ± 3.4	**91.9 ± 2.4**	82.4 ± 7.1	57.0 ± 8.3	18.8 ± 6.0	13.3 ± 5.0	9.8 ± 4.7
2.0	8.3 ± 3.4	10.9 ± 3.9	36.2 ± 6.1	41.4 ± 9.4	49.1 ± 10.6	73.1 ± 3.7	86.0 ± 2.2	**89.4 ± 2.4**	87.4 ± 2.1	48.7 ± 7.9	34.4 ± 6.8	25.6 ± 6.8
2.5	50.3 ± 9.4	50.8 ± 9.4	41.4 ± 9.8	47.4 ± 11.7	62.4 ± 10.5	82.0 ± 5.8	85.6 ± 4.4	86.2 ± 5.0	**86.4 ± 4.8**	83.3 ± 4.5	81.0 ± 4.5	78.0 ± 4.7
5.0	46.9 ± 5.6	46.8 ± 6.5	32.7 ± 6.2	42.9 ± 7.1	60.5 ± 8.2	82.7 ± 3.5	87.1 ± 2.2	89.3 ± 2.3	90.2 ± 2.2	**90.2 ± 2.5**	89.8 ± 2.7	89.4 ± 2.8
7.0	47.3 ± 6.2	45.9 ± 6.9	30.6 ± 5.9	37.9 ± 7.5	55.3 ± 8.5	80.0 ± 3.2	85.0 ± 3.0	87.1 ± 3.4	88.2 ± 3.4	89.9 ± 3.1	**89.9 ± 3.4**	89.8 ± 3.7
10.0	50.8 ± 6.3	49.6 ± 5.7	33.9 ± 7.5	39.4 ± 8.5	53.9 ± 9.3	79.6 ± 4.6	85.0 ± 4.0	86.9 ± 3.9	88.1 ± 3.5	89.3 ± 3.2	89.2 ± 3.2	**89.3 ± 3.3**

Table 9: Cross-encoding generalization (Mean ± Std %) for the RNN model. Rows indicate δ_{train}, columns indicate δ_{test}. Diagonal (in-distribution) results are bolded.

Train δ	**Test** δ											
	-1.5	**-1.0**	**-0.3**	**0.0**	**0.3**	**1.0**	**1.5**	**2.0**	**2.5**	**5.0**	**7.0**	**10.0**
-1.5	**83.5 ± 3.9**	83.7 ± 3.3	52.0 ± 7.7	23.9 ± 6.0	15.5 ± 4.4	16.2 ± 6.1	17.2 ± 5.9	18.1 ± 5.1	18.0 ± 5.5	18.3 ± 5.9	17.9 ± 6.2	17.3 ± 6.3
-1.0	88.1 ± 2.4	**90.6 ± 2.1**	75.1 ± 3.2	45.5 ± 6.6	25.4 ± 3.7	15.0 ± 3.3	14.2 ± 4.6	15.2 ± 5.5	16.9 ± 7.0	20.0 ± 6.2	21.5 ± 6.3	22.2 ± 6.2
-0.3	68.7 ± 4.8	79.8 ± 4.2	**94.1 ± 1.9**	88.0 ± 2.8	65.4 ± 4.8	14.9 ± 5.2	7.2 ± 2.3	7.6 ± 3.0	9.0 ± 3.1	17.7 ± 5.7	23.3 ± 5.1	28.7 ± 6.0
0.0	52.3 ± 5.3	65.6 ± 6.8	89.7 ± 2.7	**92.8 ± 2.8**	86.4 ± 3.7	22.1 ± 6.8	8.3 ± 3.2	6.8 ± 2.7	7.4 ± 3.4	16.9 ± 5.0	23.2 ± 7.4	29.2 ± 8.2
0.3	1.9 ± 2.5	3.3 ± 2.8	72.1 ± 4.1	91.2 ± 2.5	**93.0 ± 2.5**	67.0 ± 8.4	25.0 ± 10.2	9.3 ± 6.1	5.1 ± 4.3	3.7 ± 3.8	6.9 ± 5.5	10.6 ± 6.6
1.0	1.2 ± 1.0	2.0 ± 1.5	44.6 ± 6.8	71.9 ± 4.9	79.7 ± 3.4	**87.6 ± 3.3**	63.2 ± 4.4	14.0 ± 5.5	5.5 ± 2.8	1.8 ± 1.4	1.2 ± 1.2	0.9 ± 0.8
1.5	1.0 ± 1.3	0.9 ± 0.9	19.5 ± 5.6	36.9 ± 12.3	42.4 ± 12.6	60.5 ± 11.1	**78.2 ± 6.0**	28.5 ± 8.6	11.5 ± 6.7	4.0 ± 3.8	2.8 ± 2.8	2.3 ± 2.7
2.0	2.4 ± 2.2	2.8 ± 2.2	7.0 ± 4.4	9.5 ± 5.3	10.4 ± 8.2	18.3 ± 7.6	33.7 ± 7.2	**63.7 ± 14.1**	25.1 ± 7.6	4.9 ± 1.8	3.8 ± 1.5	2.8 ± 1.3
2.5	24.7 ± 5.3	21.5 ± 5.2	3.8 ± 2.5	0.9 ± 1.3	1.8 ± 2.1	7.4 ± 5.8	17.2 ± 9.5	25.8 ± 13.2	**26.7 ± 15.4**	23.0 ± 7.8	21.7 ± 6.0	20.0 ± 4.5
5.0	36.9 ± 10.0	33.0 ± 8.8	5.1 ± 1.5	1.2 ± 0.8	1.3 ± 1.4	7.5 ± 3.5	24.3 ± 10.6	47.4 ± 13.7	67.9 ± 12.4	**79.2 ± 7.3**	78.8 ± 6.8	77.9 ± 6.7
7.0	46.6 ± 5.6	43.4 ± 5.6	6.9 ± 2.9	1.4 ± 1.0	1.1 ± 0.8	6.2 ± 2.9	25.9 ± 7.4	55.6 ± 8.7	75.5 ± 3.1	85.9 ± 2.5	**86.1 ± 3.1**	86.1 ± 3.1
10.0	42.7 ± 6.5	39.8 ± 6.1	6.6 ± 3.2	0.8 ± 0.6	0.7 ± 0.7	3.5 ± 2.8	14.4 ± 7.6	40.4 ± 10.8	62.1 ± 12.6	83.6 ± 4.4	85.8 ± 3.3	**86.7 ± 3.1**

Table 10: Correlation between layer-wise CV and OOD accuracy across all train-test pairs (n=1440)

Layer	**Pearson's r**	**p-value**	**Significance**
Layer 1	-0.783	4.96e-299	***
Layer 2	-0.602	1.88e-142	***
Layer 3	-0.215	1.66e-16	***

Note: CV = Coefficient of Variation (σ/μ). Negative correlations indicate that lower CV (more stable firing) is associated with higher OOD accuracy. Significance levels: *** p < 0.001, ** p < 0.01, * p < 0.05, n.s. = not significant.

Table 11: Comparison of layer-wise CV across dynamical regimes.

Training Regime	**Layer**	**Mean CV** ($\mu \pm \sigma$)
Expansive ($\delta < 0$)	Layer 1	1.214 ± 0.062
($n = 240$)	Layer 2	0.716 ± 0.114
	Layer 3	$\mathbf{0.470 \pm 0.288}$
Critical / Transition ($\delta \in [0, 5]$)	Layer 1	0.723 ± 0.071
($n = 960$)	Layer 2	0.524 ± 0.055
	Layer 3	0.377 ± 0.084
Dissipative ($\delta > 5$)	Layer 1	0.673 ± 0.036
($n = 240$)	Layer 2	0.475 ± 0.026
	Layer 3	$\mathbf{0.307 \pm 0.051}$

A.2 Experiment 2: Unsupervised Receptive Field Learning Details

Dataset and Setup. The model for this experiment is a spiking neural network (SNN) autoencoder. Its architecture consists of an encoder linear layer ($W_{\text{enc}} \in \mathbb{R}^{128 \times d}$), a recurrent Leaky Integrate-and-Fire (LIF) hidden layer with 128 learnable thresholds and membrane decay $\beta = 0.95$ neurons[6], and a decoder linear layer ($W_{\text{dec}} \in \mathbb{R}^{d \times 128}$). The sparse autoencoder minimizes a composite loss balancing reconstruction fidelity and code sparsity, inspired by the sparse coding hypothesis of visual cortex[8, 9]:

$$\mathcal{L} = ||\mathbf{x}_{\text{target}} - \mathbf{x}_{\text{recon}}||_2^2 + \lambda ||\mathbf{z}_{\text{spikes}}||_1$$

where $\mathbf{z}_{\text{spikes}} = \sum_{t=1}^{T} S(t)$ is the total spike count vector. We quantify receptive field (RF) organization and training dynamics via three metrics: *Reconstruction Loss* ($||\mathbf{x}_{\text{target}} - \mathbf{x}_{\text{recon}}||_2^2$), *Sparsity Loss* ($||\mathbf{z}||_1$), and our primary metric for structural organization, *RF Standard Deviation* (σ_{RF}).

Metric Definition: $\sigma_{\text{RF}} = \sqrt{\frac{1}{N}\sum(D_{ij} - \bar{D})^2}$ quantifies the variance of learned weights, serving as a proxy for feature differentiation (structured filters have high variance, noise has low variance), where D is the learned dictionary W_{enc} and $\bar{D}$ is the global mean weight.

The learned receptive fields correspond to the rows of W_{enc}. The decoder reconstructs the static image from the temporal sum of spikes: $\mathbf{x}_{\text{recon}} = W_{\text{dec}} \sum_{t=1}^{T} S(t)$. For temporally encoded inputs, the network processes the signal sequentially.

The dataset consists of 5,000 random 16×16 grayscale patches extracted from CIFAR-10 images (converted from 32×32 RGB via standard luminance formula)[10]. Patches are normalized to the [0,1] range. We systematically compare eight input encoding strategies, which are grouped by their temporal characteristics[11, 12]:

- *Baseline (static)*: $\mathbf{x}_{\text{in}} = \mathbf{x}_{\text{target}}$. The static patch is fed at $t = 0$.
- *Random (temporal jitter)*: Each pixel's value is perturbed by $\mathcal{N}(0, 0.1)$ noise across T steps.
- *Linear (rate coding)*: Pixel intensity x_i linearly modulates firing rate, $r_i(t) = x_i \cdot t/T$.
- *Poisson (stochastic spiking)*: Pixels generate Poisson spike trains with rate $\lambda_i = x_i \cdot f_{\text{max}}$.
- *Dynamic Expansive ($\delta = -1.5$)*; *Critical ($\delta = 0.0$)*; *Dissipative ($\delta = 10.0$)*; *Transition ($\delta = 2.0$)*

Training uses the Adam optimizer[7] with a learning rate of 1×10^{-3} using a batch size of 64.

Robustness Ablations. To validate the robustness of the findings in Table 1 and ensure the structural emergence is not an artifact of specific hyperparameters, we design two follow-up validations on:

1. *Sparsity Weight (λ)*: We extended the evaluation to include $\lambda = 0$ (no explicit sparsity constraint) alongside $\lambda \in \{0.1, 0.5, 1.0, 3.0\}$. As detailed in Table 12, the Transition regime ($\delta = 2.0$) uniquely demonstrates spontaneous structural emergence even at $\lambda = 0$ ($\sigma_{\text{RF}} \approx 0.177$), whereas other regimes collapse into unstructured noise ($\sigma_{\text{RF}} < 0.05$). Across all non-zero weights, the Transition regime consistently yields the highest Structural Score.
2. *Resolution*: We repeated experiments on STL-10 (96×96 patches)[13]. Table 13 confirms the findings hold for different dataset and higher-resolution images.

Fine-Grained Landscape Validation. To rigorously decouple the effects of physical dynamics from explicit regularization, we conducted a high-resolution parameter sweep over the dynamical control parameter $\delta \in [-1.0, 5.0]$ under varying sparsity weights (λ). Figure 3 reveals in-depth interaction patterns between dynamical complexity and learnability.

While conventional methods suggest that explicit sparsity constraints (e.g., L1 regularization) are necessary to induce structured receptive fields, the most significant finding is revealed by the progression from $\lambda = 1.0$ (Fig. 3a) to $\lambda = 0.0$ (Fig. 3c): the structural quality of the learned features actually improves as the explicit constraint is removed. As shown in Fig. 3, the peak structural score increases from ~ 0.47 ($\lambda = 1.0$) to ~ 0.61 ($\lambda = 0.0$). This demonstrates that the *Transition* Dynamics ($\delta \approx 2.5$) do not merely "support" feature learning; they act as an advantageous, intrinsic constraint that is more effective than artificial regularization. This finding provides robust evidence for the biological plausibility of our model, suggesting that biological systems, operating without explicit global loss functions, can rely on metabolic and physical constraints to shape connectivity.

Furthermore, we observe a local minimum at the precise center of the critical regime ($\delta \approx 2.6$), characterized by a sharp spike in Reconstruction Loss (blue curve) coincident with a dip in RF Structure (red curve). This phenomenon likely reflects a trade-off between dynamical complexity and linear decodability: at $\delta \approx 2.6$, the system generates trajectories that are topologically rich but highly non-linear. The linear decoder struggles to reconstruct these complex signals (high loss), temporarily disrupting feature condensation. Consequently, the optimal zones for representation learning naturally settle at the *margins* of this critical peak ($\delta \approx 2$ and 3), where the system balances rich dynamical entropy with sufficient regularity for decoding.

Spectral Analysis and Scale-Space Invariance To investigate the physical mechanism underlying the structural emergence, we performed a spectral analysis of the dynamical trajectories generated by the encoder. We estimated the Power Spectral Density (PSD) using Welch's method and extracted two key metrics: the *Spectral Centroid*, representing the weighted mean frequency of the signal, and the *Spectral Entropy*, quantifying the complexity of the power distribution.

Scale-Space Invariance (Fig. 4A). To verify that these spectral properties are intrinsic to the dynamical regime rather than artifacts of specific hyperparameters, we performed a multi-scale sweep across observation scales (N) and physical evolution times ($T_{max} \in \{4, 8, 12, 16\}$). This analysis reveals a persistent spectral signature localized strictly within the transition regime, characterized by the stable coexistence of low dominant frequencies and high spectral entropy. Furthermore, this signature remains robust across varying timescales, confirming that transition dynamics actively suppress the accumulation of chaotic divergence. The sharp contrast between this ordered state and the high-frequency stochasticity of the expansive regime establishing it as a robust computational spot for generalization capacity.

Spectral Properties (Fig. 4B). We analyzed the *Spectral Entropy* and *Dominant Frequency* across the dynamical spectrum. *Expansive* regime ($\delta < 0$) is characterized by maximal entropy and high frequency, resembling broadband

Table 12: Robustness of receptive field organization across varying sparsity weights (λ). *Dynamic Transition* ($\delta = 2.0$) regime consistently produces the highest structure (RF Std) across all weights, demonstrating that the emergence of structure is robust to changes in the sparsity objective. Bold values indicate the optimal result for each metric within each λ group.

Sparsity Weight (λ)	**Encoder**	**Structure (RF Std)**	**Reconstruction Loss**	**Sparsity Loss**
$\lambda = 0.0$	Baseline	0.049821	0.003512	0.002510
(No Constraint)	Random	0.042503	0.003621	0.001905
	Linear	0.054112	0.003498	0.002285
	Poisson	0.041550	0.003905	**0.000008**
	Dynamic Dissipative ($\delta = 10.0$)	0.047015	0.003840	0.000812
	Dynamic Critical ($\delta = 0.0$)	0.036880	0.003865	0.001055
	Dynamic Transition ($\delta = 2.0$)	**0.176904**	0.003710	0.000920
	Dynamic Expansive ($\delta = -1.5$)	0.046233	**0.001275**	1.850210
$\lambda = 0.1$	Baseline	0.048941	0.002277	0.001549
	Random	0.042101	0.002528	0.001410
	Linear	0.071794	0.002119	0.001752
	Poisson	0.043657	0.003882	**0.000069**
	Dynamic Dissipative ($\delta = 10.0$)	0.063901	0.002914	0.000763
	Dynamic Critical ($\delta = 0.0$)	0.041037	0.003009	0.001006
	Dynamic Transition ($\delta = 2.0$)	**0.292495**	0.002719	0.001107
	Dynamic Expansive ($\delta = -1.5$)	0.040490	**0.001279**	0.184287
$\lambda = 0.5$	Baseline	0.043694	0.003216	0.002427
	Random	0.035226	0.003396	0.001973
	Linear	0.057604	0.003072	0.002518
	Poisson	0.046205	0.003908	**0.000006**
	Dynamic Dissipative ($\delta = 10.0$)	0.053488	0.003751	0.000578
	Dynamic Critical ($\delta = 0.0$)	0.037390	0.003769	0.000664
	Dynamic Transition ($\delta = 2.0$)	**0.236729**	0.003429	0.001379
	Dynamic Expansive ($\delta = -1.5$)	0.042407	**0.001278**	0.860399
$\lambda = 1.0$	Baseline	0.041741	0.003534	0.002529
	Random	0.034058	0.003656	0.001896
	Linear	0.054055	0.003506	0.002270
	Poisson	0.049168	0.003908	**0.000008**
	Dynamic Dissipative ($\delta = 10.0$)	0.050199	0.003858	0.000804
	Dynamic Critical ($\delta = 0.0$)	0.036652	0.003872	0.001046
	Dynamic Transition ($\delta = 2.0$)	**0.205833**	0.003722	0.000868
	Dynamic Expansive ($\delta = -1.5$)	0.043464	**0.001276**	1.716117
$\lambda = 3.0$	Baseline	0.039928	0.003777	0.002443
	Random	0.034596	0.003790	0.002142
	Linear	0.050401	0.003767	0.002183
	Poisson	0.044609	0.003907	**0.000019**
	Dynamic Dissipative ($\delta = 10.0$)	0.047220	0.003895	0.000209
	Dynamic Critical ($\delta = 0.0$)	0.036943	0.003923	0.001738
	Dynamic Transition ($\delta = 2.0$)	**0.202009**	0.003870	0.000640
	Dynamic Expansive ($\delta = -1.5$)	0.043081	**0.001255**	5.003095

Table 13: Robustness validation on the STL-10 dataset (96×96 patches) using $\lambda = 0.1$. *Transition* regime persists superior structural organization on higher-resolution natural images.

Encoder	**Structure (RF Std)**	**Reconstruction Loss**	**Sparsity Loss**
Dynamic Transition	**0.1892**	0.0051	0.0011
Dynamic Dissipative	0.0433	0.0055	0.0010
Baseline	0.0381	0.0049	0.0031
Dynamic Expansive	0.0390	**0.0024**	1.8920
Poisson	0.0415	0.0058	**0.0001**

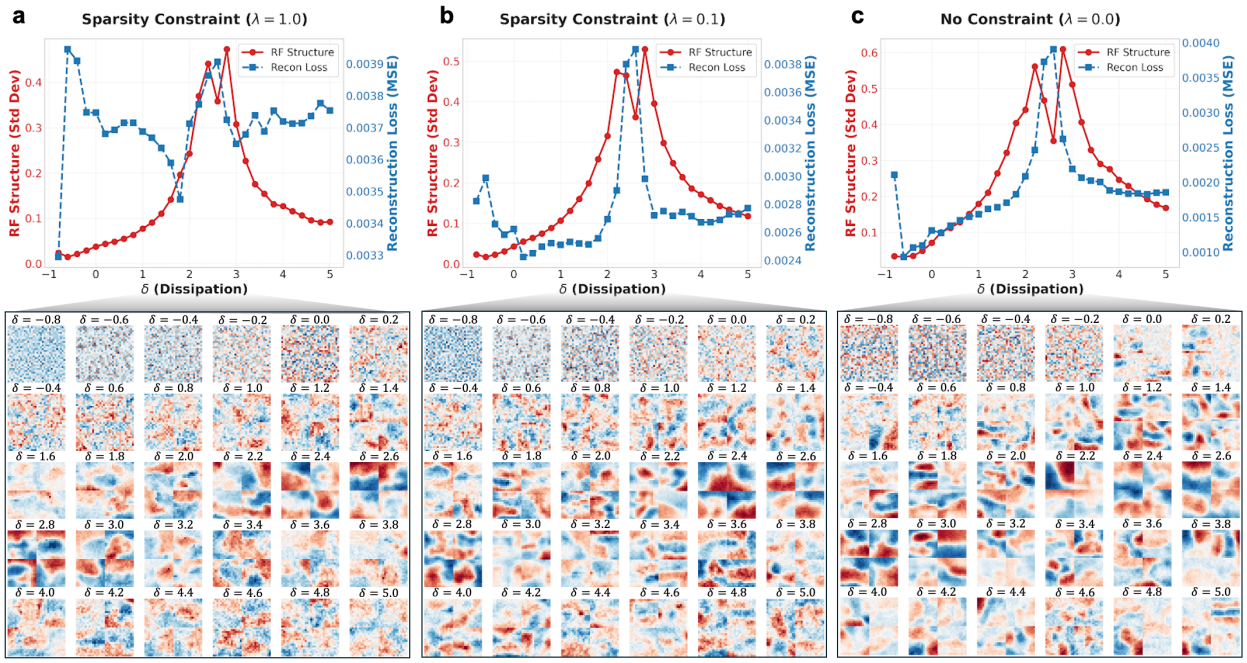

Figure 3: **Intrinsic Dynamics and the Emergence of Structure.** We analyze the impact of dynamical constraints (δ) on feature organization across three sparsity regimes: **(a)** Standard constraint ($\lambda = 1.0$), **(b)** Weak constraint ($\lambda = 0.1$), and **(c)** No constraint ($\lambda = 0.0$). While a "structural ridge" ($\delta \in [1, 3]$) persists across all regimes, the peak structural organization *increases* as the explicit sparsity constraint decreases (from ~ 0.47 at $\lambda = 1.0$ to ~ 0.61 at $\lambda = 0.0$). This trend is visually corroborated by the receptive field visualizations (bottom panels), where the diversity and clarity of structured filters visibly expand as the external constraint is relaxed. This demonstrates that the transition dynamics alone serve as a novel intrinsic inductive bias compared to external regularization.

white noise, which is difficult for neural networks to regularize. *Dissipative* regime ($\delta \gg 2$) exhibits minimal entropy, indicating information loss. *Transition* regime ($\delta \in [0, 2]$) represents a critical state: it creates a signal that is structurally complex (intermediate-to-high entropy) yet temporally structured (locked to low frequencies).

A.3 Experiment 3: Behavioral Robustness

This appendix provides detailed experimental setups and supplementary analyses for the behavioral robustness experiments described in Section 2.4. We present two complementary validation approaches: encoding-level (using Duffing transformation) and architecture-level (varying membrane leak parameter β).

A.3.1 Encoding-Level Experiments

Setup. Agents were trained exclusively in an "Easy" environment (pole length 0.5m, gravity $9.8\ m/s^2$) based on the *CartPole-v1* task [14]. They were evaluated zero-shot on three physical properties to create progressively challenging environments while maintaining the same state-space structure, as presented in Table 14.

Table 14: `CartPole` difficulty configurations. Environmental stochasticity increases through pole length, pole mass, force noise, and initial range.

Difficulty	Pole Length (m)	Pole Mass (kg)	Force Noise	Init Range
Easy (training)	0.5	0.1	0.0	0.05
Medium	0.8 (+60%)	0.3 (+200%)	0.005	0.08
Hard	1.2 (+140%)	0.5 (+400%)	0.01	0.10
Very Hard	1.5 (+200%)	0.7 (+600%)	0.015	0.12

These perturbations alter the system's moment of inertia and responsiveness without changing the observation space or action set, providing a test of learned policies' ability to generalize across physical parameter shifts[15].

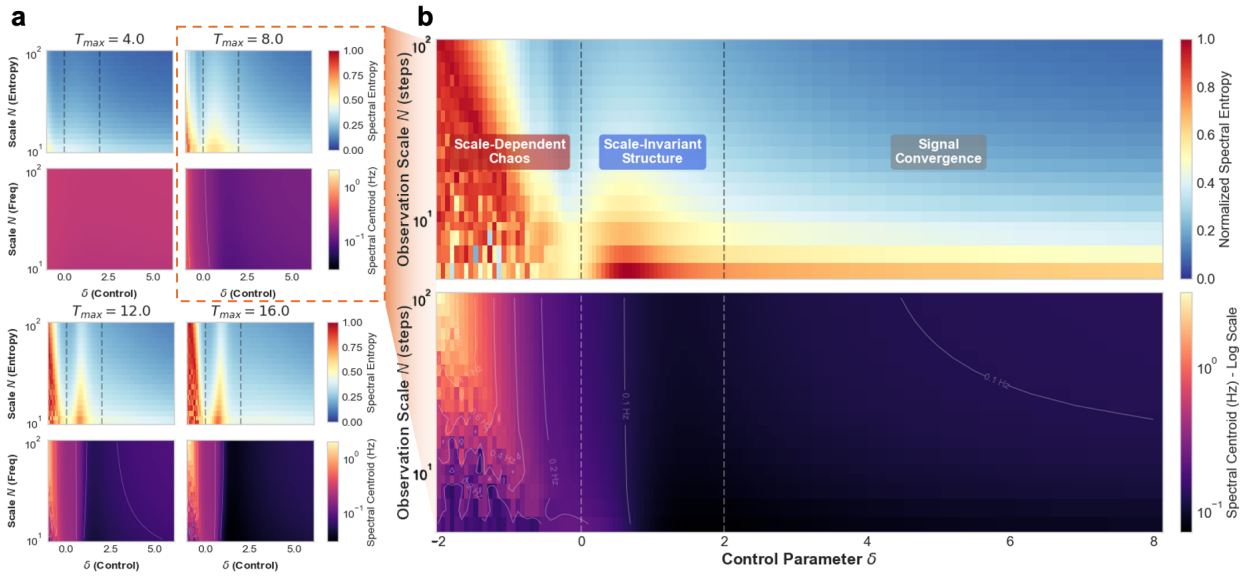

Figure 4: **Mechanism of Spectral Alignment and Invariance. (a)** Scale-space heatmaps across varying physical timescales (T_{max}) demonstrate the robustness of this mechanism. The dark vertical band in the transition region indicates that the "Low-Frequency, High-Structure" valley is *scale-invariant*, persisting stably across different observation windows. This confirms that the generalization capability arises from locking onto intrinsic invariants rather than transient artifacts. **(b)** Spectral analysis reveals a unique signature in the *Transition* regime ($\delta \approx 2$): it minimizes the *Frequency Centroid* (bottom) to align with the network's spectral bias, while simultaneously maintaining high *Spectral Entropy* (top) to preserve structural complexity.

All policy networks (SNN and MLP) shared an identical fully-connected topology of two hidden layers with 128 units each. This width was chosen to ensure sufficient network capacity, guaranteeing that performance differences would arise from the quality of the input encoding and architectural synergy, rather than from an under-parameterized model bottleneck. We use the REINFORCE algorithm[16] with Adam optimizer ($lr = 10^{-3}$, $\gamma = 0.99$)[7]. All results are averaged over 10 independent runs.

For the *Baseline MLP* agent, this resulted in a ($4 \to 128 \to 128 \to 2$) architecture processing the raw 4-dimensional environment state. For SNN and encoded-MLP agents, each of the four state variables was used to initialize a 3D oscillator mentioned in sections above, creating a 12-dimensional input vector (4 states $\times$ 3 dims), resulting in a ($12 \to 128 \to 128 \to 2$) topology. The SNN policy network operated with a LIF decay rate of $\beta = 0.95$. The MLP models used ReLU activations.

To ensure rigorous comparison, we incorporated *Layer Normalization* within the MLP hidden layers[17]. This design choice was driven by our preliminary ablations: without extrinsic normalization, the stateless MLP suffered rapid numerical divergence when processing the high-variance *Expansive* ($\delta = -1.5$) encoding. Furthermore, while the *Transition* and *Dissipative* regimes achieved comparable mean performance without normalization, they exhibited significantly amplified inter-trial variance. We therefore standardized on Layer Normalization to eliminate optimization instability as a confounder, ensuring that performance differences reflect genuine representational capabilities rather than gradient issues. In contrast, the SNN required no such extrinsic stabilization; its intrinsic leaky-integration mechanism ($\beta = 0.95$) provides natural temporal smoothing and dynamic stability.

Training followed two distinct protocols corresponding to the results in Table 2:

- *Fixed-Budget (2000 Eps.):* Agents were trained for a strict, fixed budget of 2,000 episodes (`early_stopping` = `False`). This protocol ensures all agents are compared using identical training resources, isolating the impact of encoding and architecture on generalization efficiency.

- *Sufficient Training:* Agents were trained for a maximum of 5,000 episodes with an evaluation-based early stopping mechanism (`early_stopping` = `True`). During training, the agent's policy was evaluated every 20 episodes. A policy was considered 'solved' and training was stopped if its mean evaluation reward met or exceeded 195.0 for five consecutive evaluation intervals. This protocol compares the final, converged generalization capability of each agent.

A.3.2 Architecture-Level Experiment Setup

To validate that the constraint principle generalizes beyond specific dynamical encoding, we conducted a β-sweeps where agents received raw state histories without any external dynamical transformation. This isolates the architectural contribution of the SNN's intrinsic temporal dynamics.

Setup. We tested on two tasks of increasing complexity. For `CartPole`, we used the same difficulty progression as the encoding-level experiments (Table 14). For `LunarLander-v3`, we introduced environmental stochasticity through wind, turbulence, and modified landing constraints (Table 15).

Table 15: `LunarLander` difficulty configurations. Environmental stochasticity increases through wind forces, turbulence, initial velocity perturbations, and narrower landing zones.

Difficulty	Gravity Scale	Wind Power	Turbulence	Init Vel Scale	Landing Zone
Easy (training)	0.8	0.0	0.0	0.3	2.0×
Medium	0.9	3.0	0.3	0.6	1.5×
Hard	1.0	6.0	0.7	0.9	1.2×
Very Hard	1.1	8.0	1.0	1.2	0.9×

All networks shared a two-layer topology with 256 hidden units. For baseline comparisons, we implemented:

- *MLP*: Feedforward network with LayerNorm and ReLU activations, processing single-step observations.
- *LSTM*: Single-layer LSTM (256 units) with LayerNorm, processing sequential observations with hidden state persistence across timesteps.

For SNN architectures, we implemented two variants to disentangle the effects of membrane dynamics and recurrent connectivity:

- *Leaky SNN*: Feedforward LIF network where temporal integration occurs solely through membrane potential dynamics. The membrane update follows $\text{mem}_{t+1} = \beta \cdot \text{mem}_t + W \cdot x_t$, with surrogate gradient (fast sigmoid, slope=25) for backpropagation through spikes.
- *RLeaky SNN*: Recurrent LIF network with additional lateral connections within each layer. The membrane update becomes $\text{mem}_{t+1} = \beta \cdot \text{mem}_t + W_{\text{ff}} \cdot x_t + W_{\text{rec}} \cdot s_t$, where s_t denotes the spike output. This provides an additional memory pathway beyond membrane potential.

Weight initialization followed orthogonal initialization with gain $\sqrt{2}$ for hidden layers and gain 0.01 for output layers. For RLeaky networks, recurrent weights were initialized with spectral radius 0.9 to ensure stable dynamics.

The choice of RL algorithm was driven by task complexity. For `CartPole` (4D state, 200-step episodes), vanilla REINFORCE [16] provided sufficient learning signal. For `LunarLander` (8D state, 1000-step episodes), the higher-dimensional state space and longer episodes required PPO [18] with Generalized Advantage Estimation (GAE, $\lambda = 0.95$) for stable training. Critically, neither algorithm introduces confounding temporal mechanisms: both remain policy gradient methods without learned world models, ensuring that performance differences reflect architectural properties rather than algorithmic artifacts.

Table 16 summarizes the hyperparameters for each task. SNN agents were allocated more training episodes to account for slower initial learning, while maintaining identical evaluation protocols.

After training, each agent was evaluated for 100 episodes on each difficulty level without exploration noise. We report mean reward, standard deviation, and success rate (fraction of episodes achieving reward ≥ 195 for CartPole, ≥ 200 for LunarLander). The generalization gap is defined as $\text{Gap} = \text{Reward}_{\text{Easy}} - \text{Reward}_{\text{Very Hard}}$.

Constraint versus Memory. A central question in interpreting our results is whether the observed generalization benefits stem from *temporal constraint* (phase space contraction through dissipation) or simply from *temporal memory* (information retention capacity). This distinction is critical: if generalization arises from memory accumulation, then architectures with greater memory capacity should exhibit monotonically improved robustness as retention increases. Conversely, if constraint drives generalization, we expect a non-monotonic relationship where excessive information retention (high β) permits overfitting to transient features.

RLeaky SNNs provide a critical test case to disentangle these mechanisms. Unlike feedforward Leaky SNNs, which rely solely on membrane potential for temporal integration ($\text{mem}_{t+1} = \beta \cdot \text{mem}_t + W_{\text{in}} x_t$), RLeaky SNNs incorporate lateral recurrent connections that provide an additional memory pathway through persistent hidden states ($\text{mem}_{t+1} = \beta \cdot \text{mem}_t + W_{\text{in}} x_t + W_{\text{rec}} s_t$). This architecture increases the network's capacity to retain information across timesteps[19].

Table 16: Training configurations for architecture-level experiments.

Parameter	CartPole (REINFORCE)	LunarLander (PPO)
Learning rate (actor)	5×10^{-4}	3×10^{-4}
Learning rate (critic)	—	1×10^{-3}
Gradient clipping	1.0	0.5
Hidden dimension	256	256
Max episodes (MLP/LSTM)	3,000	15,000
Max episodes (SNN)	6,000	15,000
Convergence threshold	195.0	200.0
Batch size	—	2048
PPO epochs	—	10
Clip epsilon	—	0.2

Table 17 presents complete β-sweep results for RLeaky SNNs across both tasks, with ANN and LSTM baselines for reference. Figures 5 and 6 visualize the relationship between β, generalization gap, learning dynamics, and training stability.

Table 17: RLeaky SNN results. Recurrent variants exhibit higher sensitivity to hyperparameter extremes compared to feedforward SNNs. Low or High β values (≤ 0.3 or ≥ 0.9) lead to training instability (failed convergence).

Task	Architecture	Easy Reward	V. Hard Reward	Gap ↓	Conv. Eps.
CartPole	*ANN (baseline)*	199.0 ± 2.0	60.3 ± 50.4	138.7	100
	LSTM (baseline)	169.0 ± 60.5	127.0 ± 65.2	41.9	50
	RLeaky SNN ($\beta = 0.1$)	$8.3 \pm 0.1^{\dagger}$	15.3 ± 0.2	—	1050
	RLeaky SNN ($\beta = 0.3$)	196.3 ± 2.7	171.7 ± 6.3	24.6	2250
	RLeaky SNN ($\beta = 0.5$)	198.1 ± 1.5	$\mathbf{183.8 \pm 9.7}$	**14.3**	1100
	RLeaky SNN ($\beta = 0.7$)	199.4 ± 1.0	182.9 ± 8.9	16.5	750
	RLeaky SNN ($\beta = 0.9$)	$65.5 \pm 71.1^{\dagger}$	68.8 ± 63.7	—	2050
	RLeaky SNN ($\beta = 1.0$)	$23.3 \pm 19.1^{\dagger}$	29.2 ± 16.4	—	1350
LunarLander	*ANN (baseline)*	238.9 ± 11.2	58.6 ± 40.5	180.3	400
	LSTM (baseline)	220.0 ± 14.7	26.9 ± 27.8	193.1	200
	RLeaky SNN ($\beta = 0.1$)	$21.7 \pm 292.2^{\dagger}$	-17.1 ± 145.4	—	2400
	RLeaky SNN ($\beta = 0.3$)	241.0 ± 18.6	86.0 ± 38.7	155.0	1600
	RLeaky SNN ($\beta = 0.5$)	222.8 ± 27.2	$\mathbf{110.2 \pm 33.7}$	**112.6**	1600
	RLeaky SNN ($\beta = 0.7$)	240.6 ± 14.1	89.6 ± 47.3	151.0	1000
	RLeaky SNN ($\beta = 0.9$)	$179.6 \pm 95.5^{*}$	13.1 ± 80.1	166.5	800
	RLeaky SNN ($\beta = 1.0$)	$-80.5 \pm 6.8^{\dagger}$	-169.5 ± 11.0	—	3200

†Training failed to converge. Gap not meaningful. *High variance/Partial convergence.

The comparison between Leaky and RLeaky reveals several important patterns:

- *Sensitivity and Stability Window.* While both architectures achieve their respective optimal generalization at a similar characteristic timescale ($\beta \approx 0.5$), they differ fundamentally in stability. Feedforward Leaky SNNs maintain robust performance across a broader range of regimes ($\beta \in [0.5, 1.0]$), achieving a superior generalization gap of **96.5** on `LunarLander`. In contrast, RLeaky SNNs exhibit a significantly narrower stability window: they require strict dissipative constraints to function, and their performance degrades rapidly outside the optimum (Gap: 112.6 at $\beta = 0.5$, deteriorating to training failure at $\beta = 1.0$). This suggests that the additional memory introduced by recurrent connections acts as an expansive force; to prevent chaotic divergence, the network must rely heavily on the dissipative β constraint to maintain a stable computational regime.
- *The Cost of Generalization.* Figure 6**(c, f)** reveals a stark contrast in the temporal structure of learning. Baseline models (ANN, LSTM) converge rapidly (within 400 episodes) but plateau at suboptimal generalization levels, symptomatic of "shortcut learning" that fits transient statistics. In contrast, robust SNN agents (particularly at $\beta = 0.5 - 0.7$) exhibit a much slower, gradual learning curve (requiring 1000+ episodes). This empirically illustrates the "cost of abstraction": the extraction of invariant features is inherently more computationally demanding than the memorization of surface statistics.
- *Inefficacy of Unconstrained Memory.* RLeaky SNNs possess greater theoretical memory capacity than Leaky SNNs, yet they achieve *worse* generalization (Gap 112.6 vs. 96.5 on `LunarLander`, see Table **??**) and higher

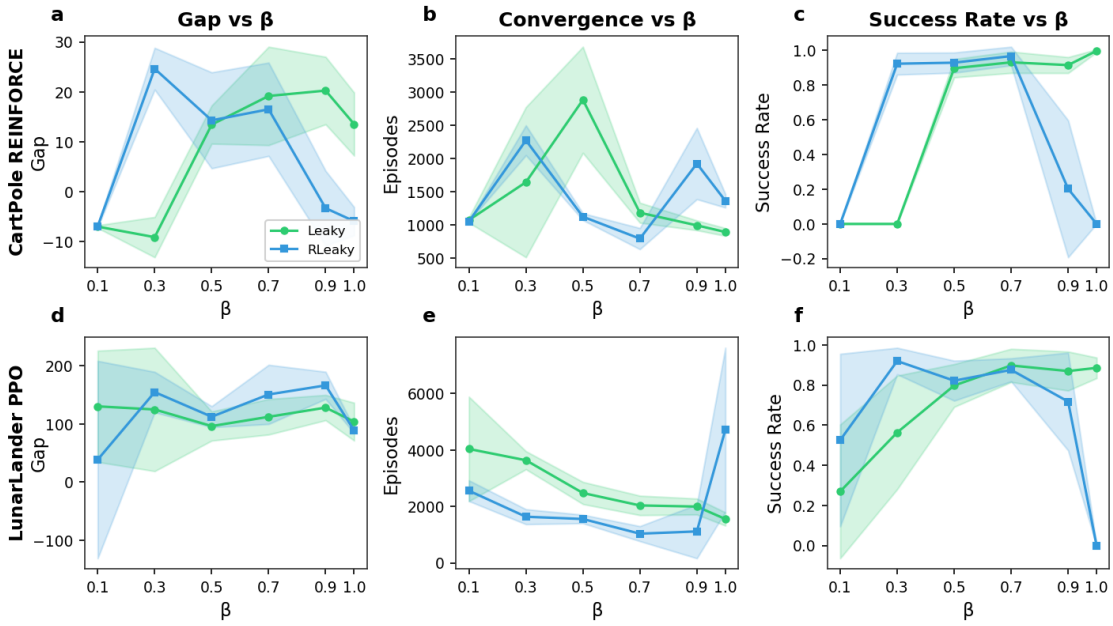

Figure 5: β-sweep comparison between Leaky and RLeaky SNNs. **Top row**: `CartPole` (REINFORCE). **Bottom row**: `LunarLander` (PPO). **(a, d)** Generalization gap versus β. Both architectures reveal non-monotonic optima. Note that the apparent low gap for RLeaky at $\beta = 1.0$ on LunarLander (d) is an artifact of training failure (see f). **(b, e)** Episodes to convergence. Higher β generally accelerates learning, but extreme values trigger instability. **(c, f)** Success rate on Easy environment. Leaky SNNs are robust across $\beta \geq 0.5$, whereas RLeaky SNNs exhibit significant instability at both extremes, indicating that recurrent memory pathways require stronger dissipative constraints to remain stable.

variance (wider confidence intervals in Figure 6). This indicates that simply adding retention pathways does not confer robustness. Instead, excess memory without sufficient constraint increases optimization difficulty, leading to erratic learning trajectories.

These non-monotonic relationships between β and generalization results across different architectures support our central claim: constraint induces invariance, not memory accumulation. Invariant representations emerge through gradual consolidation under constraints, whereas unconstrained learning leads to fragile, context-specific solutions. The additional recurrent pathway in RLeaky SNNs expands the hypothesis space without providing additional constraints, thereby increasing optimization difficulty (visible as training instability and variance) without improving generalization. This mechanism operates independently of memory capacity and represents a distinct computational principle from information retention.

A.4 PAC-Bayesian Analysis of Dynamical Regimes

Setup. To quantitatively link dynamical regimes to generalization guarantees, we performed a PAC-Bayesian analysis using a Bayesian spiking neural network (Bayesian SNN) trained on the same dataset and setting as in Experiment 1. The dissipation parameter δ spans the key regimes:

$$\delta \in \{-1.5, -1.0, 0.0, 1.0, 2.0, 5.0, 10.0\},$$

covering expansive, transition, and strongly dissipative dynamics (see Table 4). We train a two-layer Bayesian SNN with a LIF decay rate of $\beta = 0.95$. The weight matrices of both layers are equipped with Gaussian variational posteriors,

$$w \sim q(w) = \mathcal{N}(\mu, \sigma_q^2), \quad \log \sigma_q^2 \text{ learned},$$

and are optimized via the reparameterization trick[20, 21]. The network integrates spikes over $T = 5$ simulation steps to produce logits, trained with cross-entropy loss. Training uses the Adam optimizer (lr $= 10^{-3}$), batch size 32, and 100 epochs per regime.

Dynamics-induced prior. To connect the input dynamics to the prior in the PAC-Bayesian framework, we define for each regime a zero-mean Gaussian prior

$$p_\delta(w) = \mathcal{N}\big(0, \sigma_p^2(\delta)\big),$$

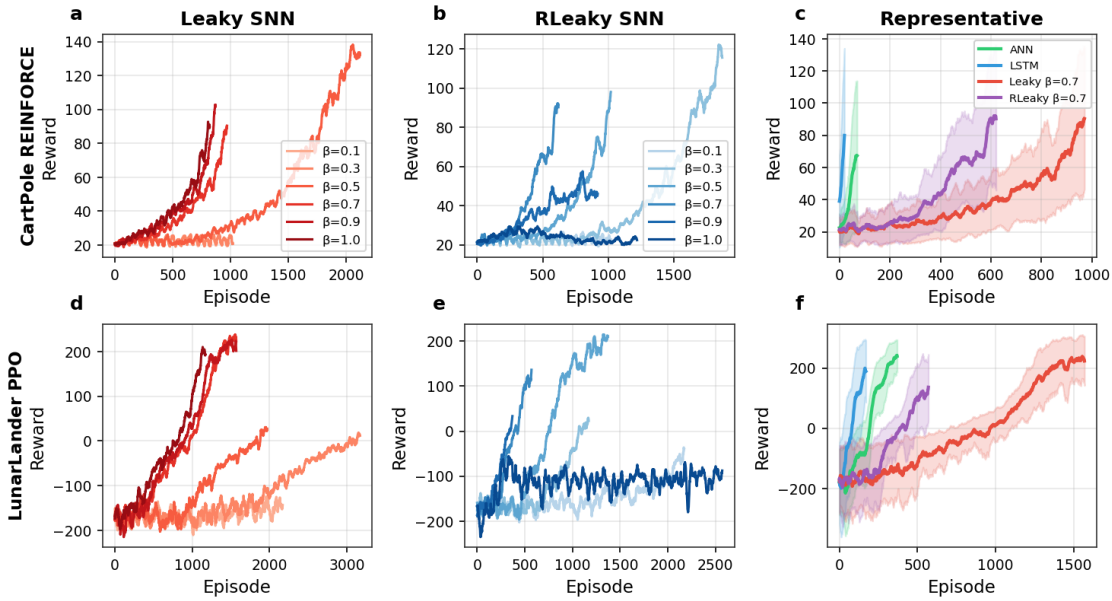

Figure 6: Training dynamics across architectures and tasks. **Top row (a–c)**: `CartPole` with REINFORCE. **Bottom row (d–f)**: `LunarLander` with PPO. **(a, d)** Leaky SNN training curves sorted by β: stronger retention (higher β, darker colors) consistently accelerates convergence speed. **(b, e)** RLeaky SNN training curves exhibit higher variance and instability, particularly at extreme β values on LunarLander. **(c, f)** Representative model comparison with confidence intervals. Baselines (ANN, LSTM) converge rapidly but generalize poorly (shortcut learning). In contrast, the robust Leaky $\beta = 0.5$ agent (red) shows a slower, more gradual learning curve, empirically illustrating the cost of abstraction: slow learning associated with robustness.

where the prior variance is tied to the global Lyapunov sum $\Sigma_\lambda(\delta)$ of the encoder through

$$\sigma_p^2(\delta) = \sigma_0^2 \exp\big(\Sigma_\lambda(\delta)\, T_{\text{enc}}\big),$$

with baseline variance $\sigma_0^2 = 1.0$. Here, $T_{\text{enc}} = 4.0$ denotes the physical evolution time of the dynamical encoding (consistent with Experiment 1). Expansive dynamics ($\delta < 0$) yield $\Sigma_\lambda > 0$ and thus a diffuse prior (large σ_p^2), whereas dissipative dynamics ($\delta > 0$) yield $\Sigma_\lambda < 0$ and a concentrated prior (small σ_p^2). For numerical stability, we clip the exponent to $[-5, 5]$, which preserves the relative ordering of regimes while avoiding degenerate variances.

PAC-Bayesian bound and metrics. For each regime, we estimate the empirical training and test errors,

$$\hat{L}_{\text{train}} = \frac{1}{m_{\text{train}}} \sum\nolimits_i \mathbb{1}[\hat{y}_i \neq y_i], \quad \hat{L}_{\text{test}} = \frac{1}{m_{\text{test}}} \sum\nolimits_j \mathbb{1}[\hat{y}_j \neq y_j],$$

where m_{train} and m_{test} are the number of training and testing samples, respectively. We define the *generalization gap* as

$$\text{Gap} = \hat{L}_{\text{test}} - \hat{L}_{\text{train}}.$$

We further compute the Kullback–Leibler divergence between posterior and prior,

$$\text{KL}(q\|p_\delta) = \sum_k \text{KL}\Big(\mathcal{N}(\mu_k, \sigma_{q,k}^2) \,\big\|\, \mathcal{N}(0, \sigma_p^2(\delta))\Big),$$

summing over all network parameters. Given $\hat{L}_{\text{train}}$, training set size $m = m_{\text{train}}$, and confidence parameter $\delta_{\text{conf}} = 0.05$, we evaluate the standard PAC-Bayesian upper bound[22] on the true risk $L(q)$, which holds with probability at least $1 - \delta_{\text{conf}}$:

$$L(q) \;\leq\; \hat{L}_{\text{train}} + \sqrt{\frac{\text{KL}(q\|p_\delta) + \ln\big(2\sqrt{m}/\delta_{\text{conf}}\big)}{2(m-1)}}. \tag{2}$$

Finally, to probe the stability of the optimization dynamics, we train a non-Bayesian SNN (same architecture, SGD optimizer) on the same encodings and record the gradient norm $\|\nabla\mathcal{L}\|$ per minibatch across 200 epochs. We summarize the gradient statistics by the mean μ_{grad} and the coefficient of variation

$$\text{CV}_{\text{grad}} = \frac{\sigma_{\text{grad}}}{\mu_{\text{grad}} + \varepsilon},$$

which quantifies the relative variability of gradients across the training trajectory, where $\varepsilon = 10^{-8}$ is a small constant for numerical stability.

Results. Table 18 summarizes the key quantities across dynamical regimes. As dissipation increases, the KL divergence exhibits a strong monotonic decrease from the expansive to the dissipative side:

Table 18: PAC-Bayesian analysis revealing the stability-performance trade-off. While Expansive dynamics ($\delta < 0$) achieve lower empirical error (Test Err), they incur a massive complexity penalty (KL), resulting in a loose generalization bound (PAC Bound). The Transition regime ($\delta = 2.0$, bolded) minimizes the PAC Bound, representing the optimal theoretical guarantee despite the local performance valley.

δ	Σ_λ	KL Divergence	Test Err	Gen. Gap	**PAC Bound** $\downarrow$
-1.5	$+3.0$	4.6×10^4	0.041	0.010	4.333
-1.0	$+2.0$	4.6×10^4	0.032	0.007	4.324
0.0	0.0	1.5×10^4	0.035	0.014	2.449
1.0	-2.0	2.3×10^3	0.083	0.012	1.040
2.0	$\mathbf{-4.0}$	$\mathbf{2.3 \times 10^3}$	**0.069**	**0.008**	**1.019**
5.0	-10.0	2.3×10^3	0.067	0.001	1.035
10.0	-20.0	2.4×10^3	0.054	-0.005	1.039

Spearman correlation between δ and KL is negative ($\rho = -0.679$), qualitatively confirming that stronger dissipation yields more concentrated posteriors relative to the dynamics-induced prior. Across all regimes, the PAC-Bayesian bound (2) remains a valid upper bound on the empirical gap (bound validity rate $= 100\%$ in our runs), although numerically loose, as is typical in high-dimensional settings.

The gradient analysis, presented in Table 19, reveals a complementary picture. The coefficient of variation of gradient norms follows a non-monotonic, U-shaped profile across regimes:

Table 19: Optimization stability analysis via gradient statistics. The transition regime ($\delta = 2.0$) achieves the lowest coefficient of variation ($\mathrm{CV}_{\mathrm{grad}}$), indicating the most stable training dynamics compared to the high variability in expansive regimes and the noise dominance in strongly dissipative regimes.

δ	Regime	μ_{grad}	$\mathrm{CV}_{\mathrm{grad}}$
-1.5	Expansive	0.558	2.492
-1.0	Expansive	0.537	1.957
0.0	Critical	0.554	1.093
1.0	Transition	0.721	0.916
2.0	Transition	0.732	**0.822**
10.0	Dissipative	0.655	1.132

The transition regime around $\delta = 2.0$ attains the lowest $\mathrm{CV}_{\mathrm{grad}}$, indicating the most stable training dynamics. Expansive regimes exhibit large and highly variable gradients, consistent with chaotic amplification of perturbations, while strongly dissipative regimes suppress gradients to the point where noise dominates, slightly increasing relative variability again.

Interpretation. Taken together, these results provide a PAC-Bayesian interpretation of the "performance valley" observed in the main experiments. At the representational and behavioral levels, networks driven by transition dynamics do not maximize instantaneous in-distribution accuracy; both expansive ($\delta < 0$) and strongly dissipative ($\delta \gg 0$) regimes can achieve comparable or even higher training performance in specific tasks. However, the PAC-Bayesian analysis shows that:

1. $\mathrm{KL}(q\|p_\delta)$ decreases sharply as we move from expansive to dissipative dynamics, indicating that dissipative temporal structure acts as an implicit prior that constrains the posterior toward a smaller, more stable region of parameter space.
2. Within the dissipative side, the transition regime minimizes gradient variability and achieves small generalization gaps while retaining sufficient model flexibility. It sits at a "sweet spot" where the network is neither driven into chaotic weight excursions (expansive) nor frozen into an over-contracted state (strongly dissipative).

In this sense, the transition regime is a *generalization optimum* rather than a performance optimum: it is the point where the trade-off between fit (training error) and stability (small KL and low $\mathrm{CV}_{\mathrm{grad}}$) is most favorable. From the perspective

of dynamical alignment, the edges of the spectrum ($\delta \ll 0$ and $\delta \gg 0$) support specialized computation—rapid discrimination or robust energy-efficient stabilization—while the transition regime provides the structural conditions under which learned solutions generalize across dynamical contexts. PAC-Bayesian theory thus formalizes the intuitive statement that "constraints breed generalization" by showing that the temporal constraints induced by dissipative dynamics manifest as tighter priors, more stable training dynamics, and theoretically justified generalization guarantees.

Appendix References

[1] Xia Chen. "Dynamical Alignment: A Principle for Adaptive Neural Computation". In: *arXiv preprint arXiv:2508.10064* (2025).

[2] Alan Wolf et al. "Determining Lyapunov exponents from a time series". In: *Physica D: nonlinear phenomena* 16.3 (1985), pp. 285–317.

[3] Joseph T Lizier, Mikhail Prokopenko, and Albert Y Zomaya. "Local measures of information storage in complex distributed computation". In: *Information Sciences* 208 (2012), pp. 39–54.

[4] E. Alpaydin and C. Kaynak. *Optical Recognition of Handwritten Digits*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50P49. 1998.

[5] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[6] Jason K Eshraghian et al. "Training spiking neural networks using lessons from deep learning". In: *Proceedings of the IEEE* 111.9 (2023), pp. 1016–1054.

[7] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*. 2015.

[8] Bruno A Olshausen and David J Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583 (1996), pp. 607–609.

[9] Horace B Barlow et al. "Possible principles underlying the transformation of sensory messages". In: *Sensory communication* 1.01 (1961), pp. 217–233.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[11] Daniel Auge et al. "A survey of encoding techniques for signal processing in spiking neural networks". In: *Neural Processing Letters* 53.6 (2021), pp. 4693–4710.

[12] Wenzhe Guo et al. "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems". In: *Frontiers in Neuroscience* 15 (2021), p. 638474.

[13] Adam Coates, Andrew Ng, and Honglak Lee. "An analysis of single-layer networks in unsupervised feature learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 215–223.

[14] Mark Towers et al. "Gymnasium: A standard interface for reinforcement learning environments". In: *arXiv preprint arXiv:2407.17032* (2024).

[15] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30.

[16] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3 (1992), pp. 229–256.

[17] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[18] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[19] Wolfgang Maass, Thomas Natschläger, and Henry Markram. "Real-time computing without stable states: A new framework for neural computation based on perturbations". In: *Neural computation* 14.11 (2002), pp. 2531–2560.

[20] Charles Blundell et al. "Weight uncertainty in neural network". In: *International conference on machine learning*. PMLR. 2015, pp. 1613–1622.

[21] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[22] David A McAllester. "Some pac-bayesian theorems". In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 230–234.