INTELLIGENCE FROM LEARNABLE NOVELTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Intelligence appears under different names in different fields: as data compression in statistics and machine learning, as universal computation in dynamical systems, and as adaptive behavior in agents. Each field has its own foundational theory, and the two most influential theories fail in mirror image: novelty search, seeking surprise, is transfixed by a noisy TV; the free-energy principle is content in a dark room. Both failures have a single cause: each objective treats as one quantity the information a learner can convert into knowledge and the information it never can. Here we show that the learnable part, which we call learnable novelty, yields the seemingly disparate projections of intelligence, and we give a fast, differentiable, closed-form estimator built on a reservoir computer. Used as a measure, without supervision, the estimator recovers decades of complexity classification, ranking the Turing-complete rule 110 highest among the elementary cellular automata. Used as an objective, its gradient carries a neural cellular automaton from simple dynamics into solitons, the traveling, colliding structures by which rule 110 computes, and organizes an image encoder around the ten digit classes of MNIST; no label ever enters training. Handed to a reinforcement-learning agent as an intrinsic reward, it improves on the task baseline in nine of ten environments and collapses in none. Complexity generation, abstraction, and exploration, ordinarily pursued in separate fields, thus emerge from ascent on one differentiable quantity, and the projections of intelligence gain a common footing.

1 INTRODUCTION

Few concepts are invoked across as many disciplines as intelligence, and few are theorized in as many incompatible ways. To statistics and machine learning it is extreme compression of data; to the study of complex systems it is the emergence of universal computation; in the interaction of an agent with its environment it is open-ended adaptive behavior. Each appearance has its own literature and its own objective function, and the literatures rarely meet. Here we show that these appearances may all follow from a single principle: the pursuit of learnable novelty.

Many creative processes unfold without a preset destination. Biological evolution has no fixed target, and scientific discovery often proceeds without knowing where it will lead. Both illustrate search in which the next direction cannot be specified in advance. Lehman & Stanley (2011) made this intuition operational as novelty search, which abandons objectives, rewards only behavior not seen before, and thereby escapes the deceptive local optima that trap goal-directed search. A parallel theory holds that compression is intelligence: a model that compresses a data stream better has captured more of the mechanism that generated it, and generalizes better for it (Hutter, 2005; Delétang et al., 2024). In the same spirit, the free-energy principle (Friston, 2010) holds that a competent agent minimizes its cumulative surprise, keeping its world compressed. Once these ideas are turned into optimization objectives, however, they often fail, each in its own way. A learner that maximizes novelty is captured by a noisy television screen: unpredictable forever, hence forever novel, it teaches the learner nothing (Pathak et al., 2017; Burda et al., 2019). A learner that minimizes surprise retreats into a dark room where nothing happens at all, because nothing is easier to predict (Sun & Firestone, 2020). The two pathologies are mirror images with a common cause: both objectives conflate the total novelty in the data with the structure a bounded mind can absorb.

Consider instead a learner of finite compute observing data one item at a time. Each item carries a quantum of surprise and an occasion to update the learner's model; as the stream accumulates, its regularities are internalized into the model while its irreducible randomness remains forever uncom-

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

pressed. The surprise summed over the whole course is exactly the stream's minimum description length (MDL) under that compute bound (Dawid, 1984; Blier & Ollivier, 2018; Rissanen, 1978; Grünwald, 2007; Finzi et al., 2026). The total splits in two: the program length of the best model that finite compute can fit, and the residual that this model can never reduce. The first part, the structure a bounded mind genuinely carries away from the data, is the *epiplexity* recently named by Finzi et al. (2026), and it is what we mean by learnable novelty; Section 2 states the decomposition formally. Finzi et al. (2026) compute epiplexity as a measure of data already in hand; we read the same quantity as learnable novelty and give it a fast, differentiable estimator, which turns it from a property to be measured into an objective a system can be driven to maximize. Seen through this decomposition, the information in a noisy television lies entirely in the residual, and a dark room contributes to neither part. Maximizing the learnable part alone therefore removes both pathologies at once.

What kind of system results when learnable novelty is maximized? For a bounded learner to keep extracting rich structure, the system it learns from must sit at the boundary of order and chaos: too ordered and there is nothing to learn, too chaotic and nothing is learnable. A dynamical system optimized for learnable novelty should therefore approach general-purpose computation, since only a system that can compute arbitrarily can keep producing learnable structure without bound (Wolfram, 1984; Langton, 1990; Cook, 2004). A representation optimized for learnable novelty must shed redundancy and emit as many mutually distinguishable, recoverable responses as the data support, so category-like structure should emerge without supervision. And an agent whose policy maximizes the learnable novelty of its own future is an agent whose death or stagnation terminates that extraction: it is driven to avoid termination, to preserve its capacity to act, and to keep its exchange with the environment rich. This sharpens the causal-entropic (Wissner-Gross & Freer, 2013) and empowerment (Klyubin et al., 2005; Salge et al., 2014) accounts of intelligent behavior: the reachable futures must be not merely diverse but learnable.

Testing these predictions means evaluating learnable novelty inside an optimization loop. The definition of epiplexity calls for a search over bounded-compute models, and the original work fully trains a neural network for every system scored (Finzi et al., 2026), a cost that becomes prohibitive during optimization. But the definition does not fix what the bounded learner is, only that it learns. A program class qualifies, a neural network qualifies, and so does a reservoir computer (Jaeger & Haas, 2004; Maass et al., 2002), in which all learnable capacity resides in a linear readout whose optimum is the closed-form solution of a ridge regression. With the reservoir as the learner, the score becomes cheap, deterministic, and differentiable in whatever produced the data: learnable novelty becomes not merely measurable but directly optimizable.

Used as a measure, it ranks the Turing-complete rule 110 (Cook, 2004) highest among the elementary cellular automata without supervision of any kind. Used as an objective, its gradient carries a neural cellular automaton (Mordvintsev et al., 2020) into a regime of complex solitons and organizes a randomly initialized MNIST (LeCun et al., 1998) encoder into digit-class clusters, with no label ever entering training. As a reinforcement-learning bonus, it improves on the task reward in nine of ten environments and collapses in none.

2 From Novelty to Epiplexity

Novelty search made the pursuit of the unseen operational but left *novelty* itself undefined: it requires a hand-picked behavior descriptor and rewards distance in that chosen space (Lehman & Stanley, 2011). To measure novelty without hand design, the most natural approach is surprise. Consider transmitting a sequence $Y = (y_1, \ldots, y_N)$ to a receiver who already holds the corresponding inputs $X = (x_1, \ldots, x_N)$. The data arrive one symbol at a time, and before each arrival the receiver predicts the symbol from what it has already seen, $p(y_i \mid y_{<i}, X)$. When y_i is revealed, the receiver pays a surprise cost $\ell_i = -\log_2 p(y_i \mid y_{<i}, X)$ and updates its predictor on the new datum. Summed over the sequence, the chain rule collapses the total cost to a single quantity,

$$L \;=\; \sum_{i=1}^{N} \ell_i \;=\; -\log_2 \prod_{i=1}^{N} p(y_i \mid y_{<i}, X) \;=\; -\log_2 p(Y \mid X), \tag{1}$$

the prequential description length (Dawid, 1984; Blier & Ollivier, 2018): the number of bits the receiver pays before X determines Y. Cumulative surprise is a natural gauge of novelty: the more

an observer who holds X is surprised in the course of observing Y, the more novel the relationship between them. Rewarding L in place of distance in a hand-chosen behavior space preserves the drive toward the unseen while removing the hand-designed descriptor.

Read this way, two established theories of intelligence occupy the two extremes of one quantity. Novelty search maximizes L: high cumulative surprise marks behavior not seen before, and chasing it outperforms goal-directed search on deceptive problems (Lehman & Stanley, 2011). The free-energy principle minimizes L: low cumulative surprise marks an internal model that fits its inputs, and a surprise-minimizing agent is read as self-organizing (Friston, 2010). Each fails, but for opposite reasons. A noisy television maximizes L per unit time because every frame is independent random noise, so a surprise-seeking agent parks in front of it and learns nothing (Pathak et al., 2017; Burda et al., 2019). A dark room minimizes L because every input repeats the last, so a surprise-minimizing agent prefers it to any structured environment (Sun & Firestone, 2020). The two failures share a root: L is a sum of two distinct components, the surprise that can be learned and the surprise that cannot. An objective that sees only the sum can be driven to either extreme by moving one component alone.

The minimum description length principle makes the two components precise (Rissanen, 1978; Grünwald, 2007; Solomonoff, 1964). Among all models that could carry Y to a receiver holding X, the minimizing one splits the transmission into a description of the model and the residual the model cannot account for,

$$L \;\approx\; \min_{M \in \mathcal{M}} \Big[\, |M| - \log_2 p(Y \mid X, M) \,\Big], \tag{2}$$

where the approximation errs only in lower-order terms (Grünwald, 2007). The first term, the program length $|M|$, is the part of the relationship an observer can *learn* and reuse on future inputs. The second is the part that, at the minimum, is unlearnable: not because no model could compress it further, but because any model that tried would only lengthen the total transmission. This decomposition identifies the flaw in each of the two earlier programs. Novelty search is dominated in the extreme by the residual term: the noisy television contributes nothing to $|M|$. The free-energy principle, in minimizing prediction error, compresses $|M|$ along with it, and is therefore drawn to the dark room. Both flaws point to the same conclusion: the quantity truly worth pursuing is $|M|$ alone.

Equation 2 is uncomputable in general, since no search can visit every program in $\mathcal{M}$, and real observers hold only finite compute; we therefore consider only a finite set of halting programs $\mathcal{M}_\phi$, given by a fixed bounded observer ϕ. The MDL under a computational bound can then be defined as

$$\begin{aligned} L^\phi(Y \mid X) &= \min_{M \in \mathcal{M}_\phi} \Big[\, |M| - \log_2 p(Y \mid X, M) \,\Big], \\ M^*_\phi(Y \mid X) &= \arg\min_{M \in \mathcal{M}_\phi} \Big[\, |M| - \log_2 p(Y \mid X, M) \,\Big]. \end{aligned} \tag{3}$$

Here we define $|M^*_\phi|$ as the learnable novelty; it is precisely the *epiplexity* of Finzi et al. (2026), the structure a finite-compute learner can actually extract from the data:

$$S^\phi(Y \mid X) \;=\; |M^*_\phi(Y \mid X)|. \tag{4}$$

By contrast, novelty search must postulate its behavior space by hand, whereas learnable novelty needs only a bounded observer to decide what counts as novel. It is this novelty under a bounded observer that makes both the noisy television and the dark room contribute zero to $|M^*_\phi|$.

Epiplexity measures data that are given; we read the same quantity as learnable novelty, a function of novelty, and give it a closed-form estimator in the next section, opening practical possibilities beyond measurement. Maximizing S^ϕ over the dynamics of a system pushes it between order and chaos (Langton, 1990; Packard, 1988), where some systems can sustain universal computation (Wolfram, 1984; Cook, 2004); maximizing it over a representation forces the encoder to shed redundancy, which is compression, long associated with understanding (Hutter, 2005; Delétang et al., 2024), and the code organizes into clusters; and maximizing it over the policy of an agent selects actions whose futures are rich and learnable, sharpening the causal-entropic account of intelligent behavior (Wissner-Gross & Freer, 2013). Section 4 tests all three predictions.

3 A CLOSED-FORM ESTIMATOR OF EPIPLEXITY

Evaluating epiplexity requires finding the optimum in the bounded model space $\mathcal{M}_\phi$, which amounts to fully optimizing a model for every system scored; the original epiplexity work trains a neural network to make the estimate (Finzi et al., 2026). This is not only computationally expensive but also opaque to gradients: nothing backpropagates through such a training run to the system being scored, so epiplexity evaluated this way serves as a measure but not as an objective.

Yet the definition does not prescribe a particular learner architecture; it only requires a system with bounded learning capacity. A reservoir computer (Jaeger & Haas, 2004; Maass et al., 2002) fits this requirement exactly: a fixed, randomly initialized nonlinear feature map followed by a single linear readout layer. With the reservoir as the bounded learner, finding the optimal model reduces to solving a ridge regression in closed form, making epiplexity cheap, deterministic, and fully differentiable.

Concretely, let ϕ be an untrained, fixed random nonlinear map. Applied to N inputs X, it yields a feature matrix $H = \phi(X) \in \mathbb{R}^{N\times m}$, where m is the reservoir feature dimension, with targets $Y \in \mathbb{R}^{N\times D}$ (the construction of ϕ is given in Appendix A.3). All learnable capacity resides in the linear readout matrix $W \in \mathbb{R}^{m\times D}$, so $\mathcal{M}_\phi$ can be defined as ϕ followed by every possible linear readout. In other words, W is a program defined on the reservoir ϕ, and computing the model's program length reduces to computing the description length of the linear operator W.

Under the minimum description length (MDL) framework (Rissanen, 1978; Grünwald, 2007), the optimal readout minimizes the total description length: a residual part plus a weight part $\mathcal{C}(W, \phi)$. Assuming Gaussian residual noise $Y = HW + \epsilon$, the residual part in bits equals a scaled mean squared error, and the total description length is

$$L(W) \;=\; \frac{\|Y - HW\|_F^2}{2\sigma^2 \ln 2} + \mathcal{C}(W, \phi). \tag{5}$$

For the program length we take a spectral description length based on the singular values $s_i(W)$ of the readout:

$$\mathcal{C}_{\text{spec}}(W) \;=\; \alpha \log_2 \det\big(I_m + \eta\, WW^\top\big) \;=\; \alpha \sum_i \log_2\big(1 + \eta\, s_i(W)^2\big), \tag{6}$$

where η is a resolution parameter and α is an overall scale on the description length. Since α changes neither the ranking between systems nor the direction of the gradient, all experiments fix $\alpha = 1/2$ and set $\eta = 1$, except the MNIST encoder, which uses $\eta = 30$ (Table 3). This log-determinant form follows the MDL principle that model complexity is measured by description length, matches the coding-rate objectives used in representation learning (Rissanen, 1978; Grünwald, 2007; Yu et al., 2020), and can also be derived from a hierarchical Gaussian prior (Appendix C). Its information-theoretic advantage is that scaling adds only logarithmic cost, and when readout directions coincide or are highly redundant they contribute no new independent singular value and incur only a logarithmically small additional cost, in compliance with the compression principle of algorithmic information theory.

With equation 6 as the program length, however, the total description length equation 5 has no closed-form minimizer, and solving it by iterative optimization would obstruct the gradients we need to pass through W to the system being scored. We therefore approximate with a ridge regression (Tikhonov & Arsenin, 1977); pricing weights by an L_2 penalty on MDL grounds has precedent (Hinton & van Camp, 1993; Musat, 2026), and a Taylor expansion confirms the approximation: for small W, $\log_2 \det\big(I_m + \eta\, WW^\top\big) \approx \eta\, \|W\|_F^2 / \ln 2$, so the spectral description length reduces to the quadratic penalty of the ridge, and σ^2, α, η merge into a single ridge parameter λ. The ridge's own shrinkage keeps the readout in the small-norm regime where the expansion is accurate. Before solving, we standardize each feature column and divide the target by a fixed scale u_Y posited in advance:

$$\tilde{H}_{\cdot c} = \frac{H_{\cdot c} - \mu_c}{\hat{\sigma}_c \sqrt{m}}, \qquad \tilde{Y} = \frac{Y - \mu_Y}{u_Y}, \tag{7}$$

where μ_c and $\hat{\sigma}_c$ are the empirical mean and standard deviation of feature column c, μ_Y centers the target, and u_Y sets the measurement precision (Appendix F). The optimal readout can then be written in closed form:

$$W_\lambda \;=\; \arg\min_W \|\tilde{Y} - \tilde{H}W\|_F^2 + \lambda \|W\|_F^2 \;=\; (\tilde{H}^\top \tilde{H} + \lambda I_m)^{-1} \tilde{H}^\top \tilde{Y}. \tag{8}$$

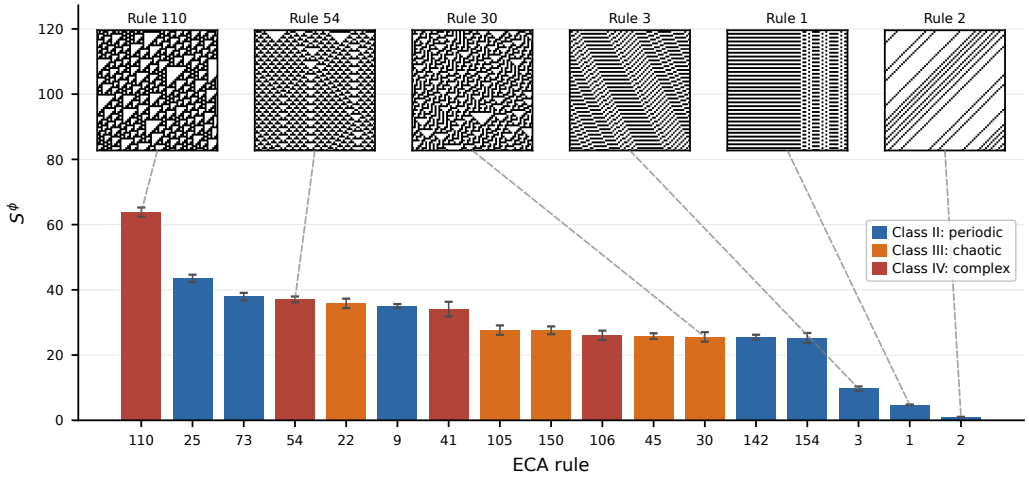


Figure 1: S^ϕ over 88 elementary cellular automata (top fourteen shown, together with the reference rules). Bars are colored by Wolfram class (Wolfram, 2002): II periodic, III chaotic, IV complex (legend); insets show the space-time diagrams (time downward) of the six reference rules 1, 2, 3, 30, 54, and 110. Each bar is the mean over ten independent draws of the random reservoir and the input ensemble, and error bars show one standard deviation. Rule 110 (Turing-complete, class IV) is the clear maximum over the whole space; at the other end, the rules whose sampled attractor dynamics are constant score exactly zero (none appears in the figure), and the near-trivial periodic rules 1 and 2 are the lowest-scoring rules shown.

Substituting W_λ into the spectral description length gives the complete reservoir-based closed-form estimator of epiplexity:

$$S^\phi(Y \mid X) = \frac{1}{2}\log_2 \det\!\left(I_m + \eta\, W_\lambda W_\lambda^\top\right) = \frac{1}{2}\sum_i \log_2\!\left(1 + \eta\, s_i(W_\lambda)^2\right), \qquad \text{where } W_\lambda \;=\; (\tilde{H}^\top \tilde{H} + \lambda I_m)^{-1}\tilde{H}^\top \tilde{Y}. \tag{9}$$

The numerical computation is detailed in Appendix E; Appendix D validates the ridge approximation against exact minimization of the description length.

4 Experiments

Intelligent behavior often appears in different guises in different systems, and we propose that these can all be understood as the product of maximizing learnable novelty. We therefore test, in turn, nonlinear dynamical systems, representation learning, and reinforcement-learning tasks. Architectures, sampling procedures, and optimization settings for all experiments are included in Appendix A.

4.1 Dynamical systems

The 256 elementary cellular automata (ECA) are one-dimensional binary lattices with nearest-neighbor update rules. Despite being describable in 8 bits each, they range from trivial to chaotic to complex; rule 110 is the only one currently proven Turing-complete (Cook, 2004).

For the discrete, finite-space elementary cellular automata we perform an exhaustive evaluation. After removing the various symmetries, we select 88 independent rules; for each we sample stationary states as the initial condition X on a width-64 ring and stack the next 32 one-step evolutions into the target Y, scoring the map with equation 9 through a convolutional reservoir. Rule 110 ranks highest in the entire rule space, the lowest scores, exactly zero, go to the rules whose sampled attractor dynamics are constant, the near-trivial periodic rules 1 and 2 score low, and the chaotic rule 30 lies in between, below the complex rule 54 (Figure 1). As a pure measurement tool, the closed-form

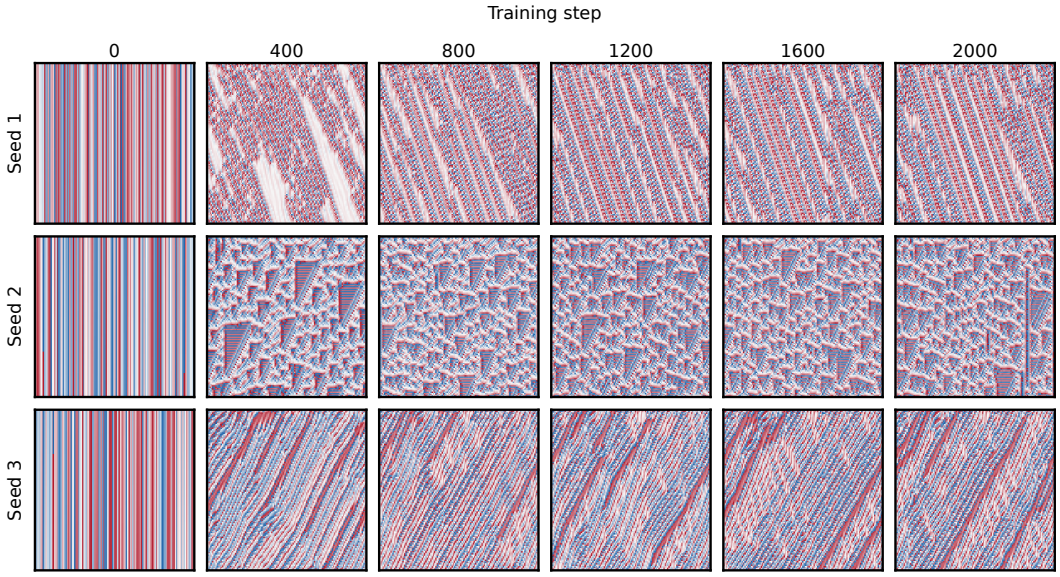


Figure 2: Inverse design of a one-dimensional NCA by gradient ascent on epiplexity. Space–time diagrams (time downward) for three seeds, one column per training step. Every seed develops solitons that travel at fixed velocity and interact on collision.

estimator reproduces the classical complexity classification without any supervision, and rule 110's top rank is robust to the estimator's hyperparameters (Appendix B). The same method, with the reservoir matched to the data geometry, also reproduces the complexity ordering of continuous-time dynamical systems (Appendix G), confirming the estimator's validity as a measure.

To test what happens when learnable novelty is directly optimized, we train a one-dimensional neural cellular automaton (NCA) (Mordvintsev et al., 2020) by gradient ascent on S^ϕ. The NCA applies a learnable local update G_θ on a ring (Appendix A); from a burned-in steady state X, the dynamics are rolled out for $\tau = 8$ steps and the resulting trajectory Y_θ is scored by the frozen reservoir, giving the objective $\max_\theta \; S^\phi(Y_\theta \mid X)$.

As the score rises, the initially simple rule develops solitons: localized structures that travel at fixed velocity and interact on collision (Figure 2; Appendix H). Solitons are often a hallmark of a capacity to transmit and combine information; rule 110 uses them to carry and combine information. The result is robust across seeds and architecture variants (Figure 5; Appendix H). The regime is a natural endpoint of the objective: a system capable of universal computation keeps producing structure a learner has not yet absorbed, so climbing learnable novelty pushes the dynamics toward the boundary of order and chaos, where coherent structures live. Solitons suit the bounded observer particularly well: a structure propagating at fixed velocity transforms predictably from step to step, a regularity the linear readout captures cheaply, while its collisions keep producing configurations the readout has not yet absorbed.

4.2 Representation learning

If the object being optimized is a representation map, maximizing its learnable novelty should force the encoder to shed redundancy and implicitly compress the data. We test this hypothesis on MNIST (LeCun et al., 1998). A trainable encoder E_θ maps an image $x \in X$ to a 64-dimensional feature vector $z = E_\theta(x) \in \mathbb{R}^D$, normalized to unit length, and is trained to maximize the learnable novelty of its own code against a fixed random MLP reservoir (Appendix H); no class label enters at any stage.

Here we maximize $S^\phi(Z = E_\theta(X) \mid X)$. After training with no class labels whatsoever, two-dimensional t-SNE projections (van der Maaten & Hinton, 2008) of the code at successive checkpoints show the initially entangled distribution progressively converging into clusters largely sep-

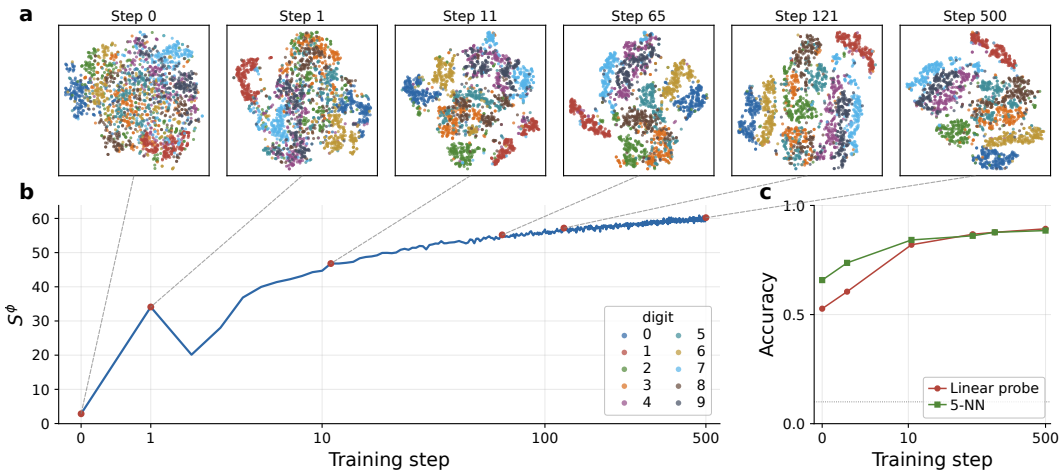


Figure 3: Unsupervised MNIST encoder trained solely to maximize reservoir epiplexity. **(a)** Two-dimensional t-SNE projections of the representation at six training checkpoints; colors indicate the held-out digit label, used for visualization only. **(b)** The S^ϕ training curve, with dashed connectors marking the checkpoint each projection is taken from. **(c)** Accuracy with which a linear probe and a 5-nearest-neighbor classifier recover the digit from the code at the same six checkpoints (chance 0.1, dotted); both rise with the epiplexity, the linear probe from 0.53 to 0.89 and the 5-nearest-neighbor from 0.66 to 0.89. Compact per-digit regions emerge progressively as the epiplexity rises, even though no class label ever enters training.

arated by digit class (Figure 3). A linear probe (Alain & Bengio, 2016) and a 5-nearest-neighbor classifier confirm this quantitatively: the accuracy with which each recovers the digit from the code rises together with the epiplexity, reaching 0.89 for both probes by the end of training (Figure 3c). The resulting representation quality is stable to moderate one-at-a-time changes of the training and estimator settings (Appendix H).

Under this objective, if the encoder merely represents highly redundant information in the data, such as the black background shared by all MNIST samples, this information barely varies across samples and contributes nothing new from the standpoint of learnable novelty. To maximize the score, the encoder must capture the features that create substantive differences between inputs. Maximizing learnable novelty is therefore implicit data compression: it forces the encoder to shed redundancy and retain only the most discriminative factors of the data. On MNIST, the dominant such factor is the digit itself, and the categorical structure of the representation space emerges spontaneously. The information bottleneck (Tishby et al., 1999) can also provide a theoretical account of this phenomenon, with the bounded observer's decodability playing the role of the relevance criterion (Appendix H.3).

4.3 Reinforcement Learning

For an agent situated in an environment, pursuing learnable novelty should steer it away from dead ends and sustain the richness of its behavior. The drive is a relative of empowerment (Klyubin et al., 2005; Salge et al., 2014), which rewards control over future states; learnable novelty asks in addition that those futures be learnable by the bounded observer. Because gradients cannot propagate through the environment, we hand the closed-form score to a standard PPO algorithm (Schulman et al., 2017; Raffin et al., 2021) as an intrinsic reward, and compare four conditions on each task: the task reward alone, the trajectory's epiplexity alone (the agent never sees the task reward), the task reward plus an epiplexity bonus, and a state-magnitude control. The per-step bonus is the increment the latest observation adds to the trajectory's epiplexity, $r_t = r_t^{\text{task}} + \beta \left(S_t^\phi - S_{t-1}^\phi \right)$, with β calibrated per environment (Appendix I).

Across ten environments spanning classic control, MuJoCo locomotion, and maze navigation, the epiplexity bonus improves on the task reward on nine tasks (Table 1). On the sparse and deceptive-

Table 1: Epiplexity as an RL reward across ten tasks (mean ± std. over ten seeds, 600,000 steps). **Bold**: improves on the task-reward baseline.

Task	Task reward	Epiplexity only	Task + magnitude	Task + epiplexity
Sparse / deceptive classic control				
Acrobot	-167 ± 166	-500 ± 0	$\mathbf{-89 \pm 4}$	$\mathbf{-83 \pm 2}$
MountainCarContinuous	28 ± 43	-97 ± 4	$\mathbf{93 \pm 1}$	$\mathbf{93 \pm 1}$
MuJoCo / Box2D locomotion				
Hopper	1879 ± 325	1006 ± 21	516 ± 128	$\mathbf{2192 \pm 270}$
BipedalWalker	125 ± 74	-59 ± 42	$\mathbf{146 \pm 51}$	$\mathbf{151 \pm 49}$
HalfCheetah	362 ± 201	-181 ± 244	$\mathbf{623 \pm 212}$	$\mathbf{393 \pm 260}$
Walker2d	296 ± 45	$\mathbf{327 \pm 45}$	294 ± 32	285 ± 41
Swimmer	181 ± 76	-12 ± 26	$\mathbf{194 \pm 76}$	$\mathbf{206 \pm 86}$
Sparse navigation				
PointMaze	229 ± 77	6 ± 3	$\mathbf{242 \pm 79}$	$\mathbf{256 \pm 22}$
Dense-reward control				
LunarLander	169 ± 74	-438 ± 393	-171 ± 35	$\mathbf{208 \pm 25}$
Pendulum	-1044 ± 42	-1092 ± 56	$\mathbf{-1016 \pm 21}$	$\mathbf{-987 \pm 23}$

reward tasks, the task reward alone leaves some seeds unsolved; with the epiplexity bonus, all seeds solve the task and the across-seed standard deviation drops sharply. It supplies the exploration drive the task reward lacks (driving the cart back and forth to build momentum, pumping the acrobot up to its bar), turning unreliable exploration into reliable solving. On the higher-dimensional locomotion tasks the bonus also lifts the return, most on Hopper.

To verify that the agent is not merely chasing large state values, we introduce state magnitude as a control reward ($\|o_t\|^2$, the squared norm of the input-normalized state), calibrated identically (Table 1). When reaching large state coincides with the task (a walker's forward position and velocity are themselves the large variables the task pays for), this crude bonus also improves on the baseline. But it is unstable: on Hopper, where large state means an extreme posture the agent cannot hold, the magnitude bonus drops far below the task reward; on LunarLander it drops harder still. The epiplexity bonus never collapses: across all ten tasks it stays at or above the task reward except on Walker2d, while the magnitude control drops below the baseline on three tasks and far below on two. The advantage of learnable novelty is that it improves performance stably across tasks. It rewards state the bounded observer can compress, whereas raw magnitude helps in one environment and hurts in another.

The epiplexity-only column shows what learnable novelty alone can do. An agent rewarded solely for its trajectory's epiplexity never sees the task and on most environments does not perform it; on Acrobot, reaching the goal would end the episode and cut off the agent's own learnable novelty, so it learns to avoid it. The exceptions are Hopper and Walker2d, where forward locomotion is itself the richest trajectory. The role of learnable novelty is therefore as a bonus that supplies the exploration the task reward lacks, promoting exploration where the reward is sparse.

The closest prior work is prediction-error curiosity (Pathak et al., 2017; Burda et al., 2019), which also rewards states an internal predictor has not yet learned. The reservoir estimator removes the inner training loop that prediction-error methods require, and the reward inherits epiplexity's immunity to noise (Section 2): a region of pure randomness inflates the agent's per-step surprise yet adds nothing to the learnable novelty, so the reward is not drawn to it.

5 Discussion

Complexity generation, abstraction, and exploration are ordinarily studied in separate fields and driven by objectives that owe nothing to one another. In the experiments reported here, all three were produced by a single quantity evaluated by a fixed observer of a single construction. Read as a measure, it recovered the classical complexity ordering of the elementary cellular automata without supervision. Ascended as an objective, its gradient carried a neural cellular automaton into a regime of solitons and organized an image encoder around the digit classes of MNIST, with no label ever entering training. Handed to an agent as an intrinsic reward, it improved on the task reward in nine

of ten environments and collapsed in none. These phenomena, we propose, are projections of one quantity, learnable novelty, onto dynamics, representations, and behavior.

What separates this account from earlier theories of intelligence is that it places the observer at its center, inside the definition itself. All three experiments position a bounded observer of the same construction at the center of dynamical evolution, representation learning, and policy execution, and the quantity being maximized is defined relative to it. The intrinsic complexity of an automaton, the quality of a code, and the adaptivity of a policy thereby cease to be absolute properties of a system and become properties of a relationship: how much structure this particular bounded observer can extract from that system. Dennett's intentional stance treats belief and agency as attributes an interpreter ascribes rather than intrinsic facts (Dennett, 1987), and Pattee's epistemic cut places the division between observer and observed within any act of measurement (Pattee, 2001); the same commitment runs through second-order cybernetics (von Foerster, 1981; Rosen, 1991). We carry this observer-centered view into a theory of intelligence. In fact, information theory has always contained an observer, but a silent one: Shannon's receiver has unbounded compute, and the observers presupposed by predictive information (Bialek et al., 2001) and computational mechanics (Crutchfield & Young, 1989) are likewise idealized, charged nothing for the compute they demand. Only recently did Finzi et al. (2026) formalize the structure available to a computationally bounded observer as epiplexity. We read the same quantity as novelty: under prequential coding (Dawid, 1984), the cumulative surprise of a learning observer splits into a learnable part and an unlearnable residual. Novelty search maximizes the sum and is dragged toward the residual; the free-energy principle minimizes it and discards structure along with noise. Learnable novelty is the part both objectives needed and neither isolates. The observer it requires is a learner: a system that receives data sequentially, pays the cost of its surprise, and updates itself with every payment. This agrees with the premise of AIXI (Hutter, 2005), where all intelligence derives from the quality of the Solomonoff predictor (Solomonoff, 1964), an idealized sequential learner whose capacity to learn sets the ceiling on the capacity to behave intelligently.

The closest predecessor of the present objective is Schmidhuber's compression progress (Schmidhuber, 2010; 2009). The two share their premise (learning is compression), but their optima part company. Compression progress rewards the rate at which the description of experience shrinks, and the total shrinkage available is set by how far that description stands above its Kolmogorov complexity. An utterly bland, unchanging stretch of experience has minimal Kolmogorov complexity and therefore the largest room to shrink, so in the long run a maximizer of cumulative progress profits most from trajectories that compress well because they contain little: this rebuilds the dark-room problem (Sun & Firestone, 2020). Learnable novelty closes this route by construction: the program a bounded observer extracts from a dark room has length near zero, and no amount of further observation makes it grow.

Two limitations bound these results. The first is that the observer never grows: freezing the reservoir buys the closed form but fixes the boundary of the learnable, and the estimator captures only the shallow structure that is linearly readable from random features. Once an optimized system exhausts what its observer can read, learnable novelty saturates. Letting observer and observed co-evolve would move that boundary rather than let the score saturate against it; open-ended growth asks for exactly such a moving boundary, and large language models may be a ready substrate, since through in-context learning a model is at once a compute-bounded sequential learner and, as a generator of sequences, the system being observed. The second limitation is that learnable novelty enters the RL experiments only as a bonus anchored by an external task reward; inverting that relationship, so that the task reward modulates where novelty is to be found, would let the agent's drive remain the pursuit of learnable novelty. Nearly all the environments tested here are finite games, and in a finite game the surest way to keep learnable novelty flowing is to avoid the terminal state; the epiplexity-only agent accordingly learns to survive and sometimes to defer the very goal that would end the episode. In settings without terminal states, such as infinite games, this framework should play a larger role.

Compression, computation, and exploration need not be assigned separate objectives: on three very different substrates, each emerged from the pursuit of learnable novelty by a bounded observer. Taking the bounded learner as the primitive, and letting observer and observed grow together, are key questions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods (writing experiment and plotting code and running the experiments), to provide feedback on research methodology and experiments through discussion of ideas, and to polish the writing of the paper. We have reviewed all AI-assisted work: code was checked and its outputs were inspected against the figures before any result entered the paper, and every AI-edited passage was read and revised by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The supplementary material contains anonymized code for every experiment, with one entry-point script per figure or table and all random seeds fixed. Dependencies are pinned so the exact package versions are installed automatically. The estimator is defined in Section 3, with the stable solve in Appendix E and normalization in Appendix F. Architectures and sampling settings are in Appendix A; inverse-experiment details in Appendix H; RL reward design and training settings in Appendix I. Robustness checks appear in Appendices B and D.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural Computation*, 13(11):2409–2463, 2001.

Léonard Blier and Yann Ollivier. The description length of deep learning models. In *Advances in Neural Information Processing Systems*, 2018.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (eds.). *Semi-Supervised Learning*. MIT Press, 2006.

Matthew Cook. Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40, 2004.

James P. Crutchfield and Karl Young. Inferring statistical complexity. *Physical Review Letters*, 63 (2):105–108, 1989.

A. Philip Dawid. Present position and potential developments: Some personal views. statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, 147(2): 278–292, 1984.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *International Conference on Learning Representations*, 2024.

Daniel C. Dennett. *The Intentional Stance*. MIT Press, Cambridge, MA, 1987.

Marc Finzi, Shikai Qiu, Yiding Jiang, Pavel Izmailov, J. Zico Kolter, and Andrew Gordon Wilson. From entropy to epiplexity: Rethinking information for computationally bounded intelligence. *arXiv preprint arXiv:2601.03220*, 2026.

Karl Friston. The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11 (2):127–138, 2010.

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pp. 5–13, 1993.

Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer, 2005.

Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

Alexander S. Klyubin, Daniel Polani, and Chrystopher L. Nehaniv. Empowerment: A universal agent-centric measure of control. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2005.

Christopher G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37, 1990.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 2020. doi: 10.23915/distill.00023.

Tiberiu Musat. Neural weight norm = Kolmogorov complexity. *arXiv preprint arXiv:2605.10878*, 2026.

Norman H. Packard. Adaptation toward the edge of chaos. In *Dynamic Patterns in Complex Systems*, pp. 293–301. World Scientific, 1988.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pp. 2778–2787, 2017.

Howard H. Pattee. The physics of symbols: bridging the epistemic cut. *BioSystems*, 60(1–3):5–21, 2001.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Robert Rosen. *Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life*. Columbia University Press, New York, 1991.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment: An introduction. In *Guided Self-Organization: Inception*, pp. 67–114. Springer, 2014.

Jürgen Schmidhuber. Ultimate cognition à la Gödel. *Cognitive Computation*, 1(2):177–193, 2009.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations*, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ray J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7(1): 1–22, 1964.

Zekun Sun and Chaz Firestone. The dark room problem. *Trends in Cognitive Sciences*, 24(5): 346–348, 2020. doi: 10.1016/j.tics.2020.02.006.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. Winston, 1977.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377, 1999.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Heinz von Foerster. *Observing Systems*. Intersystems Publications, Seaside, CA, 1981.

Alexander D. Wissner-Gross and Cameron E. Freer. Causal entropic forces. *Physical Review Letters*, 110(16):168702, 2013.

Stephen Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):1–35, 1984.

Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Table 2: Sampling and reservoir settings for the elementary cellular automata.

Sampling		*Reservoir*	
Lattice width	64	Feature map	circular 1D convolution
Boundary conditions	circular	Depth	3
Initialization	Bernoulli$(1/2)$	Channels	256
Burn-in steps	1000	Kernel size	3 (1 in the final layer)
Stacked target window	$\tau = 32$	Activation	ELU
Samples per rule N	512	Ridge λ	0.03

Table 3: Reservoir feature maps used as bounded observers. Every map places a pre-activation normalization before each nonlinearity, holding it at the edge of chaos independent of depth and architecture (Appendix A.3).

Data geometry	Feature map ϕ	Settings
1D cellular automata	circular 1D convolution	256 channels, kernel 3 (1 in the final layer), ELU; depth 3, $\lambda = 0.03$ (ECA ranking); depth 4, $\lambda = 0.3$ (inverse NCA)
Continuous-time flows	random MLP	depth 4, width 64, ELU, $\lambda = 0.1$
Flattened images (MNIST encoder)	random MLP	depth 4, width 2048, ELU, $\lambda = 3$, $\eta = 30$
RL observation trajectories	random MLP	depth 4, width 32, ELU, $\lambda = 0.3$

A Architectures and sampling

A.1 Elementary cellular automata

There are 256 elementary rules in total; after removing those equivalent under the left–right reflection and the $0 \leftrightarrow 1$ colour swap, 88 locally unique rules remain (the constant rule 0 among them), and these are the ones scored in the main text. Each elementary rule r has a binary update map F_r acting on a width-64 line with circular boundary conditions. For each rule we draw 512 independent initial states $x_i^{(0)} \sim \mathrm{Bernoulli}(1/2)$, evolve each for a 1000-step burn-in so that it lands on the rule's attractor, and take as target the stacked window of its next 32 one-step states,

$$x_i = F_r^{1000}\big(x_i^{(0)}\big), \qquad Y_i = \big(F_r(x_i),\, F_r^2(x_i),\, \ldots,\, F_r^{32}(x_i)\big),$$

where F_r^k denotes k applications of the rule. We repeat the scoring ten times per rule, each time redrawing both the random reservoir weights and the input ensemble, and report the mean and standard deviation of the resulting epiplexity. Table 2 lists these settings together with the reservoir used to score them.

A.2 Reservoir feature maps

The bounded observer is a fixed, randomly initialized reservoir whose architecture matches the data geometry. Table 3 lists the configurations used in this paper.

A.3 Reservoir criticality: a pre-activation normalization at the edge of chaos

The reservoir is random and untrained, so unlike a learned network nothing during fitting moves it toward a useful representation: whether its nonlinearity is engaged at all is decided by its initialization and by the scale of its input. A plain random feedforward stack is a poor default here. Driven by an $O(1)$ input, the pre-activations of a depth-four ELU reservoir have a standard deviation of only a few tenths, so each ELU operates near the origin where it is indistinguishable from a linear

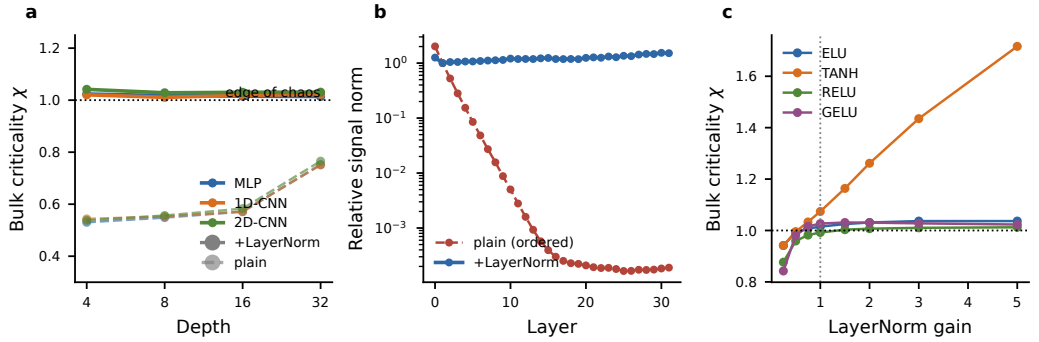


Figure 4: A pre-activation normalization holds a random reservoir at the edge of chaos. **(a)** Bulk criticality χ against depth for MLP, 1D-, and 2D-convolutional reservoirs: the plain reservoir (dashed) sits in the ordered phase, well below one, while the normalized one (solid) holds at $\chi \approx 1$ for every architecture and depth. **(b)** Signal survival through depth, the relative norm of an input perturbation by layer: the plain reservoir contracts it exponentially, the normalized one preserves it. **(c)** Bulk χ against the normalization gain for four activations; gain one reaches $\chi \approx 1$ for all of them.

map, and the feature map it computes is close to a random linear projection. The operating point is also unstable: it drifts with depth and shifts by orders of magnitude with the input scale, so one architecture can be nearly linear in one experiment and saturated in another.

For recurrent reservoirs the analogous question is settled by the echo-state property and the spectral radius. The underlying principle is that a reservoir is most expressive at the *edge of chaos*, the boundary between an ordered regime where signals contract and a chaotic one where they explode (Langton, 1990; Packard, 1988). We define the order parameter as the per-layer perturbation multiplier

$$\chi = \mathbb{E}\,\frac{\|\delta h^{(\ell+1)}\|}{\|\delta h^{(\ell)}\|}, \tag{10}$$

the factor by which an infinitesimal input perturbation grows from one layer to the next (Poole et al., 2016; Schoenholz et al., 2017): $\chi < 1$ is the ordered phase, where signal and gradients vanish with depth, $\chi > 1$ the chaotic phase, and $\chi \approx 1$ the edge of chaos. Measured by finite differences, the plain reservoir sits in the ordered phase for every architecture we test, at $\chi \approx 0.55$ for depths up to sixteen and rising to $\chi \approx 0.75$ at depth thirty-two, always well below criticality.

We find that a single rule suffices to hold the reservoir at the edge of chaos: normalize the pre-activations over the feature (channel) axis immediately before every nonlinearity. Fixing the pre-activation scale to unit variance places each nonlinearity in its responsive region and pins χ to a value the activation alone sets, independent of depth, input scale, and architecture (Figure 4).

A.4 Trainable systems

The neural cellular automaton, the MNIST encoder, and the reinforcement-learning policy are the systems trained to maximize learnable novelty. Table 4 lists their architectures and optimization settings; the reservoirs they are scored against are in Table 3.

Table 4: Trainable systems and their optimization settings.

Neural cellular automaton ($x_{t+1} = \mathrm{normalize}(x_t + g_\theta(x_t))$, circular boundary)	
Lattice width	64
Lift convolution	radius 2 (kernel 5) to 128 channels, batch normalization, GELU
Hidden layer	one pointwise convolution with batch normalization and GELU
Output projection	pointwise convolution to the two state channels
Burn-in steps	32 (no gradient tracking)
State noise	Gaussian, standard deviation 0.1, added after burn-in
Stacked target window	$\tau = 8$
Batch size	2048
State constraint	two-channel field, unit norm per site
Optimizer	AdamW with cosine annealing, learning rate 1×10^{-4}, S^ϕ scaled by $1/100$ before the gradient step
Gradient clipping	0.5
Training steps	2,000
MNIST encoder (code $z = E_\theta(x)$, no labels)	
Encoder E_θ	trainable MLP with hidden widths $64, 128, 256$ and code dimension $D = 64$
Code constraint	unit norm, $z/\|z\|$
Batch size	128
Optimizer	AdamW with cosine annealing
Training steps	500
Reinforcement-learning policy (PPO, default MLP policy)	
Algorithm	PPO (Schulman et al., 2017; Raffin et al., 2021)
Parallel environments	8
Rollout length	1024
Minibatch	256
Discount γ	0.999
GAE λ_{GAE}	0.98
Learning rate	3×10^{-4}
Training steps	600,000

For the neural cellular automaton, g_θ denotes the convolutional stack in Table 4, and $\mathrm{normalize}$ divides the two-channel vector at each lattice site by its Euclidean length, $\mathrm{normalize}(v)_i = v_i/\|v_i\|$ over the two channels at site i, so every site lies on the unit circle. The main-text run uses the residual normalized update, which keeps a skip connection before the unit-norm projection,

$$x_{t+1} = \mathrm{normalize}\big(x_t + g_\theta(x_t)\big).$$

A direct variant drops the skip connection and normalizes the convolutional output alone,

$$x_{t+1} = \mathrm{normalize}\big(g_\theta(x_t)\big).$$

For both update forms, training samples unit-norm two-channel initial states on a width-64 ring, applies 32 no-gradient burn-in steps, and adds independent Gaussian noise of standard deviation 0.1 at every site of the burned-in state; the stacked future window $Y = (G_\theta(X), \ldots, G_\theta^\tau(X))$ is rolled out and scored from this perturbed state. Without the perturbation, a run that settles onto a spatially uniform fixed point presents the reservoir with a constant input–target pair, whose score is zero and provides no gradient to escape it. Figure 5 holds the architecture, optimizer, and target window fixed and contrasts the two update rules, training each from nine independent random seeds. Both rules develop coherent traveling structures across all nine seeds.

B Robustness of the elementary-cellular-automaton ranking

The ranking of Figure 1 is computed under the fixed estimator configuration of Table 2. To test how much of it depends on that choice, we vary one hyperparameter at a time around the reference

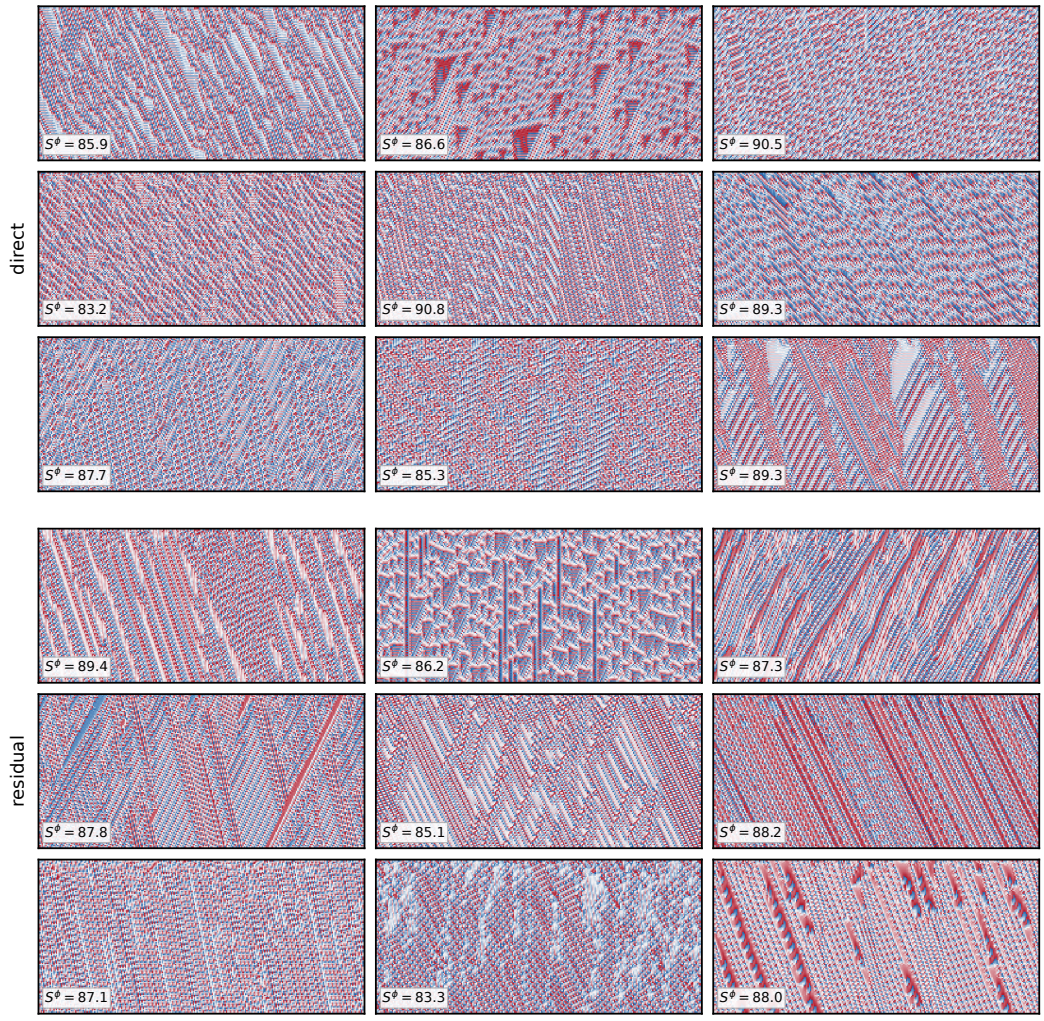


Figure 5: Final NCA evolution under the direct and residual update rules, each over nine random seeds, with the architecture and target window held fixed. The upper three rows use the direct normalized update, the lower three the residual update; the nine cells of each block are nine independent seeds. Every panel shows a final-checkpoint rollout and annotates its final S^ϕ. Both rules develop coherent traveling structures across all nine seeds, with final scores in a narrow band.

configuration and re-score all 88 rules: the resolution $\eta \in [0.03, 30]$, the ridge penalty $\lambda \in [0.001, 1]$, the target window length $\tau \in \{4, \ldots, 64\}$, the reservoir depth $\in \{1, \ldots, 5\}$, kernel size $\in \{3, 5, 7\}$, channel count $\in \{64, \ldots, 512\}$, and the sample count $N \in \{128, \ldots, 1024\}$. Each configuration is scored with a single draw of the reservoir weights and input ensemble; the reference configuration, every configuration at which the top rank changes, and every configuration at which the score order of rules 30 and 54 flips are re-scored with three independent draws.

Rule 110 keeps the top rank across the entire tested range of every axis except the two that cut the observer or the target window below the scale of its structure (Figure 6a). At the reference configuration its margin over the best other rule (rule 25) is 20.7 bits over three draws, and the margin stays between 9.6 and 28.4 bits at every other configuration where it is first. The full ranking is similarly rigid: its Spearman correlation with the reference ranking stays above 0.90 everywhere except at depth ≤ 2. The first exception is $\tau = 4$, where the window is too short for the accumulated structure of rule 110 to separate it from the field and it falls 0.5 bits behind rule 73 (18.2 ± 0.4 against 18.7 ± 0.2, three-draw means). The second is small depth: at depth 2 the reservoir's receptive field

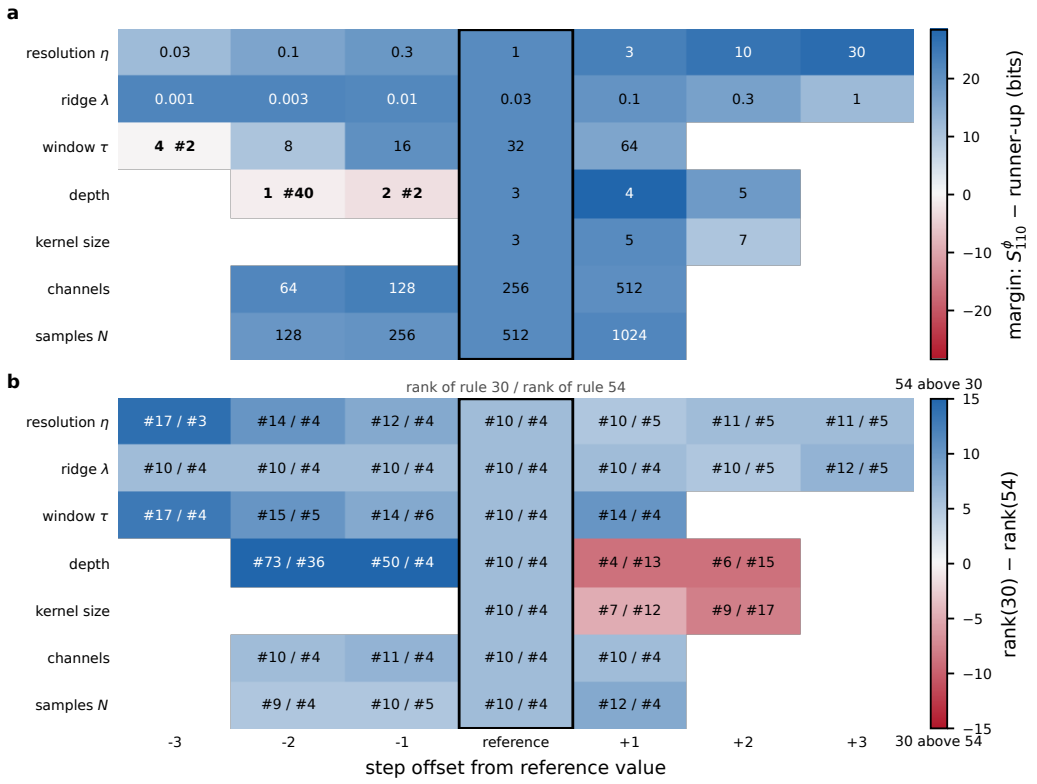


Figure 6: One-at-a-time robustness of the ECA ranking. In both panels each row varies one estimator hyperparameter around the reference configuration of Table 2 (outlined column), with columns aligned by step offset from the reference value. **(a)** Margin, in bits, between rule 110's epiplexity and that of the best other rule; each cell prints the value scored, so a positive (blue) cell is a configuration at which rule 110 ranks first, and in the cells where it is not first the cell adds rule 110's rank in bold (single draw). **(b)** Ranks of the reference rules 30 (chaotic, before the slash) and 54 (complex, after the slash) under the same scan; color gives their difference, blue where rule 54 ranks above rule 30 and red where rule 30 ranks above rule 54, saturated at ± 15.

has radius one and rule 25 overtakes rule 110 (13.5 ± 0.4 against 10.4 ± 0.5 bits), and at depth 1 the reservoir sees a single cell and the measurement collapses altogether (rule 110 falls to rank 40 and the correlation with the reference ranking vanishes).

The reference rules 30 and 54 order themselves by the observer's locality (Figure 6b). At the reference configuration the complex rule 54 ranks fourth and the chaotic rule 30 twelfth, with a score gap $S^{\phi}_{54} - S^{\phi}_{30} = 10.9 \pm 1.0$ bits over three draws, and this order holds across the full tested range of the resolution, ridge, window, channel, and sample axes. It reverses exactly where the receptive field grows beyond radius two: one more layer (depth 4, radius three) puts rule 30 fifth and rule 54 thirteenth (gap -14.6 ± 1.0 bits), and a wider kernel (kernel 5, radius four) does the same (-11.9 ± 0.5 bits). An observer of radius at most two prices the structured rule above the chaotic one; a wider view makes the chaotic rule's short-range structure learnable and reverses the order. Rule 110 ranks first in both regimes.

C The log-determinant description length as a matrix-t marginal

The main text prices the readout directly by the spectral description length equation 6. The same form arises from a hierarchical Gaussian prior. Let $W \in \mathbb{R}^{m \times D}$ and make its conditional law matrix

Gaussian,

$$W \mid \Lambda \;\sim\; \mathcal{MN}_{m,D}\big(0, \Lambda^{-1}, I_D\big), \qquad p(W \mid \Lambda) \;\propto\; |\Lambda|^{D/2} \exp\Big(-\tfrac{1}{2}\operatorname{tr}\big(\Lambda W W^\top\big)\Big), \tag{11}$$

with $\Lambda \succ 0$ a row precision matrix. Holding the precision fixed at $\Lambda = \lambda I_m$ collapses the negative log-density, in bits, to a quadratic,

$$-\log_2 p(W \mid \lambda I_m) \;=\; C + \frac{\lambda}{2\ln 2}\,\|W\|_F^2, \tag{12}$$

the ridge penalty. If instead the precision is not specified in advance but given an isotropic Wishart-type hyperprior $p(\Lambda) \propto |\Lambda|^{(a-m-1)/2} \exp\big(-\frac{1}{2\eta}\operatorname{tr}\Lambda\big)$, with $a > m-1$ so that the prior is proper and the integrals below converge, the marginal density of W is $p(W) = \int_{\Lambda\succ 0} p(W \mid \Lambda)\, p(\Lambda)\, d\Lambda$. Collecting the powers and exponential terms in Λ,

$$p(W) \;\propto\; \int_{\Lambda\succ 0} |\Lambda|^{(a+D-m-1)/2} \exp\Big[-\tfrac{1}{2}\operatorname{tr}\big((WW^\top + \eta^{-1} I_m)\Lambda\big)\Big]\, d\Lambda, \tag{13}$$

and applying the Wishart integral identity $\int_{\Lambda\succ 0} |\Lambda|^{(\nu-m-1)/2} \exp\big(-\frac{1}{2}\operatorname{tr}(A\Lambda)\big)\, d\Lambda \;\propto\; |A|^{-\nu/2}$ with $\nu = a + D$ and $A = WW^\top + \eta^{-1} I_m$ gives

$$p(W) \;\propto\; \det\big(WW^\top + \eta^{-1} I_m\big)^{-(a+D)/2} \;\propto\; \det\big(I_m + \eta\, WW^\top\big)^{-(a+D)/2}. \tag{14}$$

Dropping the constants independent of W,

$$-\log_2 p(W) \;=\; C + \frac{a+D}{2}\,\log_2 \det\big(I_m + \eta\, WW^\top\big), \tag{15}$$

which is the spectral description length equation 6 with $\alpha = (a+D)/2$. The log-determinant form is therefore the description length of the Gaussian readout after the unknown precision has been integrated out: the ridge penalty corresponds to a fixed precision, the log-determinant to a marginalized one. The marginalization supplies the *form* of the description length; the estimator treats α as a free overall scale and fixes $\alpha = 1/2$ throughout, rather than the D-dependent value the hierarchical derivation would assign.

D Exact minimization of the description length

The main text fits the readout by a ridge approximation to the total description length (Section 3). This appendix minimizes the objective directly, without the approximation:

$$\mathcal{J}_{\mathrm{MDL}}(W) \;=\; \frac{1}{2\sigma^2 \ln 2}\,\|\tilde{Y} - \tilde{H} W\|_F^2 \;+\; \mathcal{C}_{\mathrm{spec}}(W), \tag{16}$$

with $\mathcal{C}_{\mathrm{spec}}$ the spectral description length equation 6 and the residual scale calibrated as $\sigma^2 = \lambda/(2\alpha\eta)$, so that the two objectives coincide in the small-readout regime, where $\log\det(I_m + \eta\, WW^\top) \;\approx\; \eta\,\|W\|_F^2$ and the spectral description length reduces to the ridge penalty. We write S^{ϕ}_{MDL} for the spectral description length of the exact optimum W_{MDL}.

The objective has no closed-form minimizer, but the log-determinant is concave in $WW^\top$, so its linearization at the current iterate is a global upper bound on the description length, and minimizing that bound is a weighted ridge solve:

$$W_{k+1} \;=\; \big(\tilde{H}^\top \tilde{H} + \lambda\, M_k\big)^{-1} \tilde{H}^\top \tilde{Y}, \qquad M_k \;=\; \big(I_m + \eta\, W_k W_k^\top\big)^{-1}. \tag{17}$$

Each sweep of this majorize–minimize iteration decreases $\mathcal{J}_{\mathrm{MDL}}$ monotonically. Warm-started at W_λ and run in double precision on the sufficient statistics, it converges to a stationary point; rescaling the warm start by factors between $1/4$ and 4 leaves the solution unchanged.

Figure 7 compares the two readouts on identical reservoirs and data. The exact minimizer extracts more bits from the same data: across the 88 elementary rules (ten reservoir and data draws each, the setting of Table 2), S^{ϕ}_{MDL} exceeds S^{ϕ} by a factor growing from about 1 to about 2.3 with the score itself, because the log-determinant description length is flatter than the quadratic penalty at large singular values and lets the readout reach further into faintly expressed directions. The

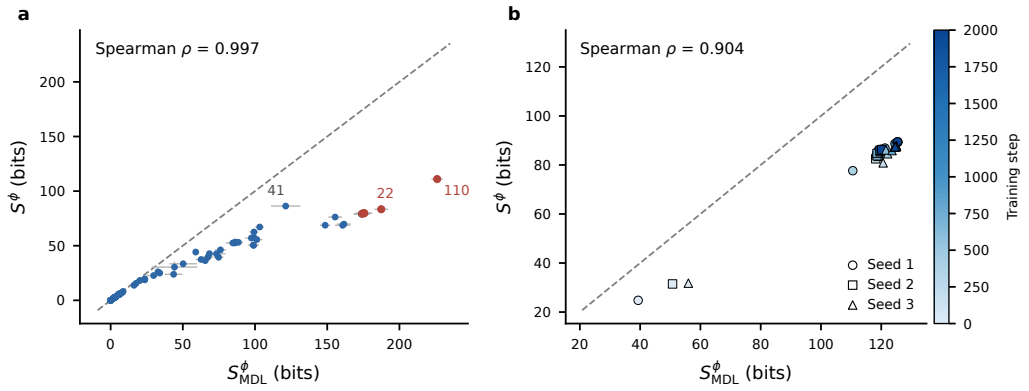


Figure 7: The ridge readout against the exact minimizer of the description length, on identical reservoirs and data. **(a)** Per-rule mean scores for the 88 elementary rules: the reported S^ϕ (the spectral description length of the ridge readout W_λ) against S^ϕ_{MDL} (that of the minimizer of equation 16). Error bars are standard deviations over ten reservoir and data draws; the dashed line is the identity; red marks the five highest rules under S^ϕ_{MDL}, and rule 41 is the largest rank change (second to thirteenth). **(b)** The same comparison at every checkpoint of the three inverse-NCA training runs (marker shape: seed; colour: training step; error bars: standard deviations over three evaluation draws).

excess is one-sided: under an isotropic feature Gram every stationary point of equation 16 satisfies $w\left(1+\lambda/(1+\eta w^2)\right) = \sigma$ along each singular direction and so shrinks less than the ridge solution $\sigma/(1+\lambda)$, giving $S^\phi \le S^\phi_{\mathrm{MDL}}$; the measured Gram is far from isotropic, but the ordering holds in every paired solve we ran (880 for the ECA, 99 for the NCA). Yet the two readouts rank systems almost identically: Spearman $\rho = 0.997$ over the rules, the same top-fourteen set, rule 110 maximal under both with its margin over the runner-up widening from 24.7 to 38.7 bits, and the largest rank change anywhere is rule 41, second under the ridge readout and thirteenth under the exact one; on the inverse-NCA trajectories (three seeds, eleven checkpoints each) the two scores move together (Pearson $r = 0.996$), with a nearly constant 35-bit offset on the converged plateau. Removing the approximation changes how many bits the observer extracts, not which systems it finds structured; the estimator therefore keeps the ridge readout.

E Numerically stable, differentiable ridge solve

The estimator in equation 9 is defined through the ridge optimum $W_\lambda = (\tilde{H}^\top \tilde{H} + \lambda I_m)^{-1} \tilde{H}^\top \tilde{Y}$. Solving it the textbook way, by forming the Gram matrix $\tilde{H}^\top \tilde{H}$ and inverting (the normal equations), squares the conditioning of the design matrix, turning a condition number $\kappa(\tilde{H})$ into $\kappa(\tilde{H})^2$. In single precision this is not a minor loss: whenever the reservoir features are nearly collinear, which is exactly the low-epiplexity regime where the bounded readout finds little independent structure to use, the squared conditioning corrupts W_λ and therefore the score. Because the inverse experiments drive the generating system across a wide range of feature conditioning, an unstable solve distorts both S^ϕ and the gradient the optimization follows.

The batch solve therefore never forms the Gram matrix. For output coordinate j, writing the ridge objective as a single least-squares problem gives

$$\left\|\tilde{H} w_j - \tilde{y}_j\right\|_2^2 + \lambda \|w_j\|_2^2 \;=\; \left\| \begin{bmatrix} \tilde{H} \\ \sqrt{\lambda}\, I_m \end{bmatrix} w_j - \begin{bmatrix} \tilde{y}_j \\ 0_m \end{bmatrix} \right\|_2^2, \tag{18}$$

where $\tilde{y}_j$ is column j of $\tilde{Y}$. The ridge optimum is the ordinary least-squares solution of the augmented system with design matrix $\tilde{H}_{\mathrm{aug}} = [\,\tilde{H};\ \sqrt{\lambda}\, I_m\,] \in \mathbb{R}^{(N+m)\times m}$ and target $\tilde{y}_{j,\mathrm{aug}} = [\,\tilde{y}_j;\ 0_m\,]$. We solve this with a reduced QR factorization $\tilde{H}_{\mathrm{aug}} = QR$ followed by a triangular

solve,

$$w_j = R^{-1} Q^\top \tilde{y}_{j,\mathrm{aug}}, \tag{19}$$

and stacking the column solutions gives $W_\lambda = [\, w_1, \ldots, w_D \,]$. This operates on the design matrix directly. The effective conditioning is that of $\tilde{H}_{\mathrm{aug}}$, the square root of what the normal equations would face, and the whole computation stays in the input dtype without ever materializing $\tilde{H}^\top \tilde{H}$. The $\sqrt{\lambda}\, I_m$ block gives $\tilde{H}_{\mathrm{aug}}$ full column rank for any $\lambda > 0$, so R is invertible and the solve is well posed even when $\tilde{H}$ is itself rank-deficient.

This solve is differentiable. Both the QR factorization and the triangular solve have well-defined derivatives, so W_λ (and hence the spectral score $S^\phi = \alpha \log_2 \det(I_m + \eta\, W_\lambda W_\lambda^\top)$, whose singular-value decomposition is likewise differentiable) is a differentiable function of $\tilde{H}$ and $\tilde{Y}$, and through the normalization and the frozen reservoir ϕ, of the data (X, Y). In the inverse experiments we backpropagate S^ϕ through exactly this computation into the parameters of the system that generates (X, Y), with ϕ held fixed. The closed-form solve therefore serves both as a cheap forward score and as a differentiable layer in the optimization.

F Normalization of features and targets

The feature normalization in equation 7 standardizes each column of the reservoir output to zero mean and unit variance; the $\sqrt{m}$ factor keeps the output scale of a random readout with i.i.d. standard normal entries invariant across feature widths. The target is divided by a fixed scale u_Y rather than by its own empirical standard deviation because the target's magnitude itself carries information: a larger-magnitude target demands a larger readout, which the spectral description length equation 6 prices at more bits, logarithmically in its scale. The constant u_Y thus acts as the measurement precision posited for the target, the unit relative to which readout magnitude and residual are priced.

G Reservoir epiplexity on continuous-time flows

The same closed-form score, with its reservoir an MLP on the state vector, ranks a further family of dynamical systems, continuous-time flows, consistently with a longstanding qualitative classification.

G.1 Continuous-time chaotic systems

For continuous-time flows we use the same random MLP reservoir as the other feedforward experiments: a four-layer network with ELU activations, hidden width 64, and ridge penalty $\lambda = 0.1$. Each sample pairs an attractor state x with a stacked window of its future: we draw a random initial condition, integrate the system for a burn-in time of 0.5 to reach the attractor, and record the state as x; Rössler and Thomas receive a longer initial integration before this burn-in (a pre-burn-in), because their attractors lie far from the region where initial conditions are drawn and the 0.5-unit burn-in alone would not reach them; then we snapshot the trajectory at the ten lead times $0.1, 0.2, \ldots, 1.0$ from x and stack them as the target window y. Both x and every target snapshot are standardized per state dimension by the empirical mean and standard deviation of the ensemble of x, so the attractor's absolute scale does not enter the score. The reservoir maps x to features and one ridge readout per snapshot predicts the whole window; the epiplexity is the program length of the stacked readout, scored as in equation 9. We draw 512 samples per system and average the score over eight independent draws of the reservoir weights.

Across three chaotic systems (Lorenz, Rössler, Thomas) and three two-dimensional linear systems (pure rotation, damped spiral, stable node), the chaotic systems score well above the linear baselines (Figure 8). Lorenz scores highest at 46, with Rössler at 25 and Thomas at 17; all three sit well above the linear systems, which fall between 6.9 and 8.4.

G.2 Temporal accumulation of epiplexity

The cross-system comparison above draws independent state pairs from an ensemble of initial conditions. In the reinforcement-learning setting, where a dense per-step reward is needed, we instead

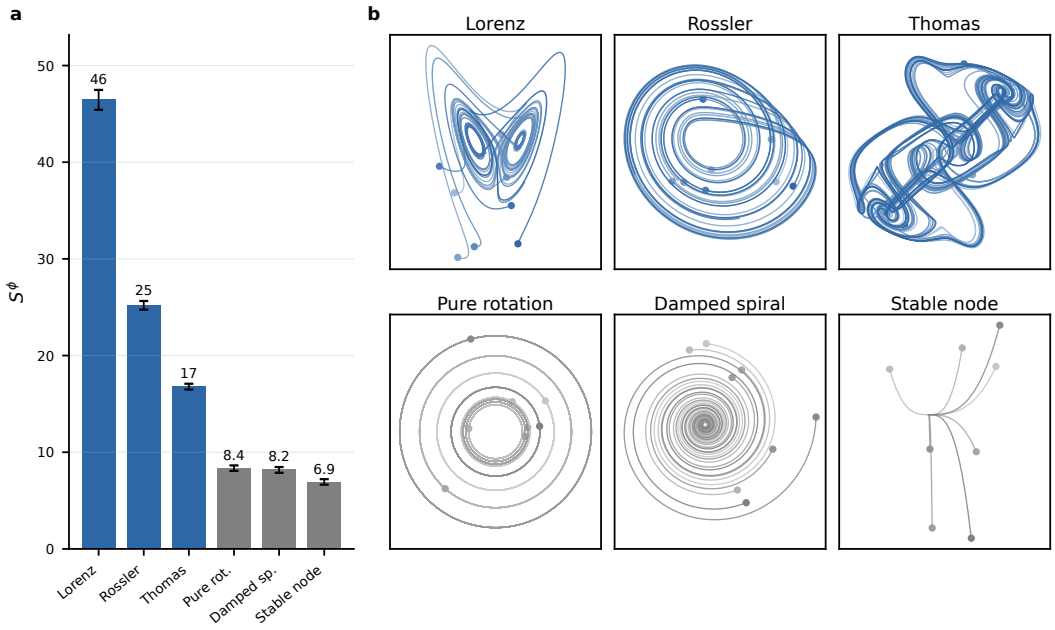


Figure 8: MLP-reservoir epiplexity on continuous-time systems. **(a)** The chaotic systems (Lorenz, Rössler, Thomas) score well above the linear baselines. **(b)** Phase portraits of the six systems, eight initial conditions each.

track how epiplexity accumulates along a single trajectory as observations arrive one by one. Given a trajectory $x_0, x_1, \ldots, x_N$ sampled at interval Δt_s, define the cumulative epiplexity S_t^ϕ as the score computed from all pairs (x_i, y_i) with $i \leq t - \tau$, where y_i is the future window stacked from step i and τ its horizon in samples. Each new observation adds one pair to the ridge regression; the marginal contribution $\xi_t = S_t^\phi - S_{t-1}^\phi$ measures how much learnable structure the latest state brings.

Recomputing the ridge readout from scratch at every step costs $O(N^2 m^2)$ for a length-N stream. The same quantity admits an exact recursive least-squares form: maintaining the inverse Gram matrix and the readout under a Sherman–Morrison rank-1 update advances each step in $O(m^2)$, plus $O(m^2 D)$ for the singular values of the $m \times D$ readout whenever the score itself is evaluated. Unlike the batch QR solve of Appendix E, this covariance form does maintain the inverse Gram matrix and so gives up that solve's conditioning advantage; at the small reservoir widths used online ($m = 32$) this is numerically benign.

H Additional details for the inverse experiments

H.1 Neural cellular automaton

With the architecture and optimization settings of Table 4, the score climbs from its low initial value over the first several hundred steps, then rises more gradually and levels onto a plateau by about step 1,500, ending between $S^\phi = 86$ and 89 across the three seeds of the residual $\tau = 8$ run shown in Figure 2 (training curves in Figure 9a). We test both the direct and residual update forms; over nine independent seeds each, both reach the same epiplexity band and develop solitons at every seed (Figure 9b, Figure 5). The traveling structures emerge early in the climb and, once formed, persist through the remainder of training. Rolling the final checkpoint out from a fresh random initial state on a wider lattice preserves the same regime beyond the training width, indicating that the learned local rule has not simply memorized the training-time lattice size.

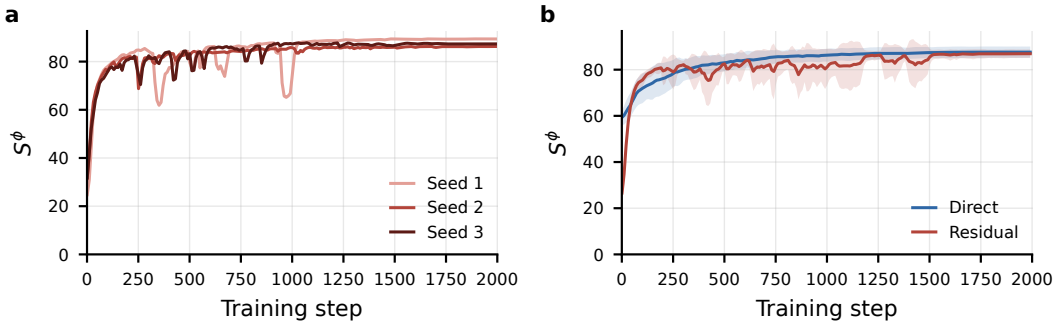


Figure 9: Training curves for the inverse NCA experiment. **(a)** S^ϕ over training for each of the three seeds shown in Figure 2. **(b)** S^ϕ for both update rules (direct and residual), each drawn as the mean $\pm$ one standard deviation over nine independent seeds; both rules reach the same epiplexity band.

H.2 MNIST ENCODER

The ridge regularizer λ controls how tight the readout bound is and therefore which encoders qualify as high-epiplexity. We use $\lambda = 3$, far above the values typical of reservoir computing (10^{-5} or smaller): a tight bound is what keeps redundancy expensive and forces the encoder toward compressed codes organized by the data's underlying factors, rather than letting it satisfy a loose readout with arbitrary ones. The raw score grows with both the code dimension D and the reservoir width, but a higher raw score reached by enlarging either does not by itself sharpen the class structure: it lets the encoder satisfy the readout without compressing. We report $D = 64$ with a width-2048 reservoir; tightening the bound, rather than enlarging the model, is what selects for natural abstractions.

We test local robustness by varying one hyperparameter at a time around the reported configuration and training each variant for 150 steps with the same seed (Figure 10). Learning rate, batch size, code dimension, and reservoir depth leave the final linear-probe accuracy between 0.80 and 0.90 across their tested ranges. The estimator parameters have a wider failure regime: very weak ridge regularization ($\lambda \leq 0.3$) or low resolution ($\eta \leq 0.1$) drives accuracy below 0.5, whereas the neighborhood containing the reported $\lambda = 3$ and $\eta = 30$ remains high-accuracy. Thus the result does not depend narrowly on the precise reference values, although the estimator must still impose a sufficiently selective readout.

H.3 INFORMATION-BOTTLENECK INTERPRETATION

The spontaneous cluster structure of the MNIST encoder can be read through the information bottleneck (Tishby et al., 1999). For an input X, a relevance variable Y, and an intermediate representation Z, the bottleneck minimizes $I(X;Z) - \beta\, I(Y;Z)$ over encoding distributions: it keeps the information in Z that bears on Y and compresses the rest. The bottleneck needs the external variable Y to decide which bits are worth keeping; in our objective that criterion is supplied implicitly by the decodability of the bounded observer: whatever structure the reservoir's short readout can recover counts as relevant. Ridge shrinkage biases this criterion toward functions of low norm in the random-feature kernel space, functions that vary smoothly along the data manifold, and among these the class is the factor carrying the most discriminative information (the cluster assumption of semi-supervised learning (Chapelle et al., 2006)); this experiment sets $\lambda = 3$, far above the other experiments (Table 3), precisely so that the relevance criterion admits only this smooth structure. The role of the compression term $I(X;Z)$ falls to the structural bottleneck: the code dimension is finite, and the spectral description length is nearly insensitive to repeated readout directions, so redundancy is squeezed out of the code.

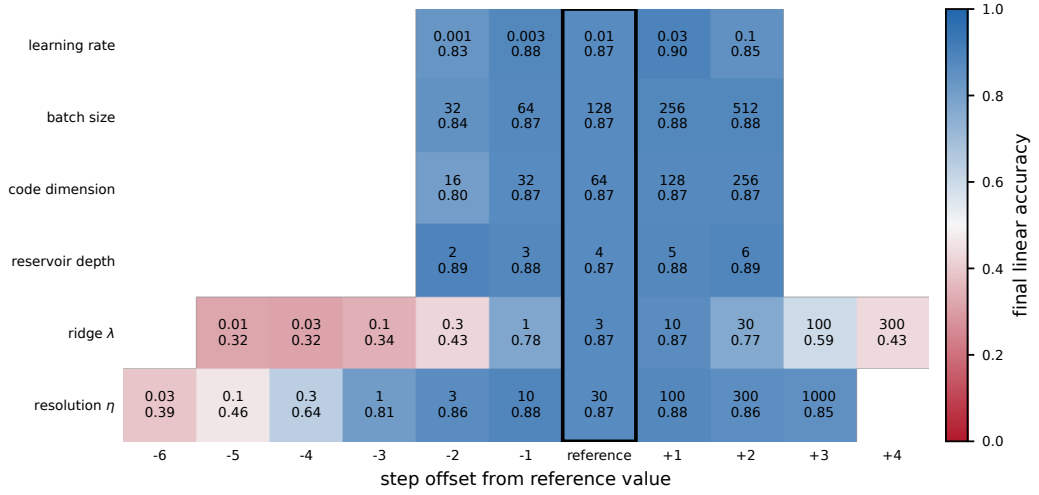


Figure 10: One-at-a-time robustness of the MNIST representation experiment. Each cell gives the varied hyperparameter (top) and final linear-probe accuracy after 150 training steps (bottom); all other settings remain at the reference values in the black-outlined column. Columns order the tested values within each row by their offset from the reference. The color scale is fixed from 0 to 1, centered at white for 0.5 accuracy; red marks values below 0.5 and blue values above it.

I Reinforcement-learning details

I.1 Reward design

From the observation trajectory $o_0, o_1, \ldots$ we score, at each step, the map from o_t to the stacked window $(o_{t+1}, \ldots, o_{t+\tau})$ of the next τ observations, flattened into a single $d\tau$-dimensional target, using the estimator of Section 3 with a frozen random four-layer MLP reservoir (ELU nonlinearities). The whole-trajectory score S_T^ϕ is maintained online by the covariance-form recursive-least-squares estimator (Appendix G.2 gives the recursion), one rank-1 update per step; the per-step reward is the increment $S_t^\phi - S_{t-1}^\phi$, which telescopes to S_T^ϕ.

I.2 Reinforcement-learning setup

Input normalization is essential here and specific to the reinforcement-learning setting (the inverse experiments use bounded states and do not need it): an agent's observation coordinates are unbounded and can differ in scale across observations by orders of magnitude, so we standardize each coordinate (subtracting its random-policy mean and dividing by its standard deviation) before the reservoir, since a large-magnitude input otherwise saturates the fixed nonlinearity in ϕ and the reward stops discriminating. A *small reservoir* (width 32) is the bounded-learner setting: a wide reservoir can fit even a random or chaotic trajectory, so the epiplexity bonus would reward chaos; a narrow one can only fit structured trajectories, so it rewards coherent, regular motion. The *window* horizon τ is a fixed per-task constant, on the order of twice the characteristic time over which the state changes appreciably under a random policy, kept within $[8, 48]$. Per task this gives (state dimension d, horizon τ): Acrobot (6, 8), MountainCarContinuous (2, 28), Hopper (11, 10), BipedalWalker (24, 40), HalfCheetah (17, 16), LunarLander (8, 48), Walker2d (17, 16), Swimmer (8, 16), Pendulum (3, 16), PointMaze (8, 48). The reservoir ridge is $\lambda = 0.3$. The PointMaze observation is the flattened goal-conditioned dictionary, and the goal resamples on contact (`continuing_task`) so reaching it does not end the episode.

I.3 Training

We use PPO (Schulman et al., 2017; Raffin et al., 2021) with 8 parallel environments, rollout length 1024, minibatch 256, $\gamma = 0.999$, GAE $\lambda_{\mathrm{GAE}} = 0.98$, learning rate 3×10^{-4}, for 600,000

steps, on the default MLP policy, reporting the mean over ten seeds. In the mixed mode the bonus weight β is calibrated per environment so the bonus's whole-episode contribution is 0.1 times the random-policy task-return scale (anchored at the episode level so it survives sparse rewards, where a per-step calibration to a near-zero reward would vanish). The state-magnitude control uses the identical setup, with the per-step bonus the squared norm $\|(o_t - \mu_o) \odot \sigma_o^{-1}\|^2$ of the input-normalized state in place of the epiplexity increment; it is run on all ten tasks of Table 1. The reported return is the task return over 100 evaluation episodes, which the epiplexity-only agent never observes during training.

On the sparse and deceptive classic-control tasks the gain is not only in the mean but in reliability: PPO on the task reward alone solves Acrobot and MountainCarContinuous only on some seeds and fails catastrophically on others (across-seed standard deviation ≈ 166 and 43 in task-return units), whereas the bonus solves them on every seed (standard deviation ≈ 2). The bonus turns unreliable exploration into reliable solving. On the locomotion tasks it lifts the mean return (Hopper most, then HalfCheetah, BipedalWalker, and Swimmer), helping PPO escape mediocre local optima. It is rarely harmful: at the same $600{,}000$-step budget the bonus also raises LunarLander's return (by 23%) and Pendulum's slightly, and only on Walker2d, where the task reward already affords adequate exploration, does it mildly lower the mean (by 4%). The pattern matches the mechanism: the bonus helps exactly when covering observed state is the bottleneck, and the learnable-novelty drive is a mild distraction when it is not.

The state-magnitude control shows that the stability of the bonus is specific to learnable novelty and not shared by raw magnitude. It collapses LunarLander from a return of 169 to -171: a near-constant coordinate (the leg-contact indicators, which a random policy almost never triggers, so their estimated standard deviation is tiny) gives the input-normalized magnitude an enormous spike whenever a leg touches down, and the agent is driven toward that spike instead of toward a soft landing. On Walker2d both the magnitude control and the epiplexity bonus sit essentially at the baseline (294 and 285 against 296), neither helping nor collapsing, and on Pendulum the magnitude control leaves the return essentially unchanged. Across all ten tasks the epiplexity bonus falls at most 4% below the task reward and collapses on none, whereas the magnitude control collapses on two (Hopper and LunarLander). Raw magnitude has no principled scale, so a single degenerate coordinate can hijack it; epiplexity instead prices each coordinate by how well a bounded readout predicts it, which is what makes it stable where the magnitude bonus is not.