NEURAL FIELDS ENCODE ADAPTATION GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural fields are usually evaluated by how well they reconstruct an observation. We show that this misses two useful properties of a fitted network: how easily it can adapt to new observations, and what its weights retain from earlier ones. We study these properties as *adaptation geometry*. For images, we meta-learn class-specific initializations, adapt each one to a new image, and measure how much the network must change to fit it. A simple local linear model closely predicts this adaptation cost, while replacing one network's tangent kernel with another's substantially worsens the prediction. Adaptation thus depends on the local geometry of the fitted network, not only on its current reconstruction. For physical fields, we repeatedly fit the same network to observations from a sequence. Its weights then retain information about that history. When two wave histories end at exactly the same observation, the final weights recover the sign of the wave velocity with 68.6% accuracy, whereas the current observation alone contains no such information and gives 50%. These two phenomena are quantitatively linked: tangent-kernel eigenvalues predict both which changes are easy to learn and how quickly they are overwritten by later fitting. Together, these results show that neural fields contain useful information beyond what they currently reconstruct: in how they can change and in how they got there.

1 INTRODUCTION

Suppose we fit a small neural network to each frame of a physical simulation. We could start every fit from the same initialization, or continue training the network fitted to the previous frame. Both procedures produce a network that reconstructs the current frame. But are the resulting networks equally useful representations? This question is natural for *neural fields*, also called implicit neural representations (INRs): networks that map coordinates to signal values, such as pixel positions to image intensities (Sitzmann et al., 2020b). Their fitted weights can be used as downstream data (Dupont et al., 2022; De Luigi et al., 2023; Navon et al., 2023), and the fitting procedure affects how useful those weights are (Papa et al., 2024). We ask what reconstruction quality alone misses.

We support this view with two main empirical results. Our contribution is not a new way to fit neural fields, but to isolate two properties of a fitted network that its current reconstruction does not reveal. First, we separate what can be predicted from the network's current output from what is gained by knowing how the network responds to further parameter updates. Second, we test whether the weights can retain information about how they were reached even when the current observation is held exactly fixed (Figure 1).

Local geometry predicts adaptation. In our image experiments, we learn separate initializations for each class and ask which one can be adapted most easily to a new image. A local linear model of how each network responds to small weight changes predicts the relative cost of these nonlinear adaptations and which initialization is preferred. Swapping this local geometry between classes consistently worsens those predictions, even after controlling for its overall scale (Section 4.1).

Sequential fitting creates useful memory. Why should history help when the underlying physical system is Markovian? A single observed field need not determine the relevant dynamics: state may be missing or reduced, or the dynamics themselves may be unknown. The Mori–Zwanzig formalism makes the first case precise: the exact dynamics of a reduced set of observables contain a memory term that integrates over their own past (Mori, 1965; Zwanzig, 1961), and this is why explicit memory helps learned PDE surrogates under partial observation (Buitrago Ruiz et al., 2025).

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

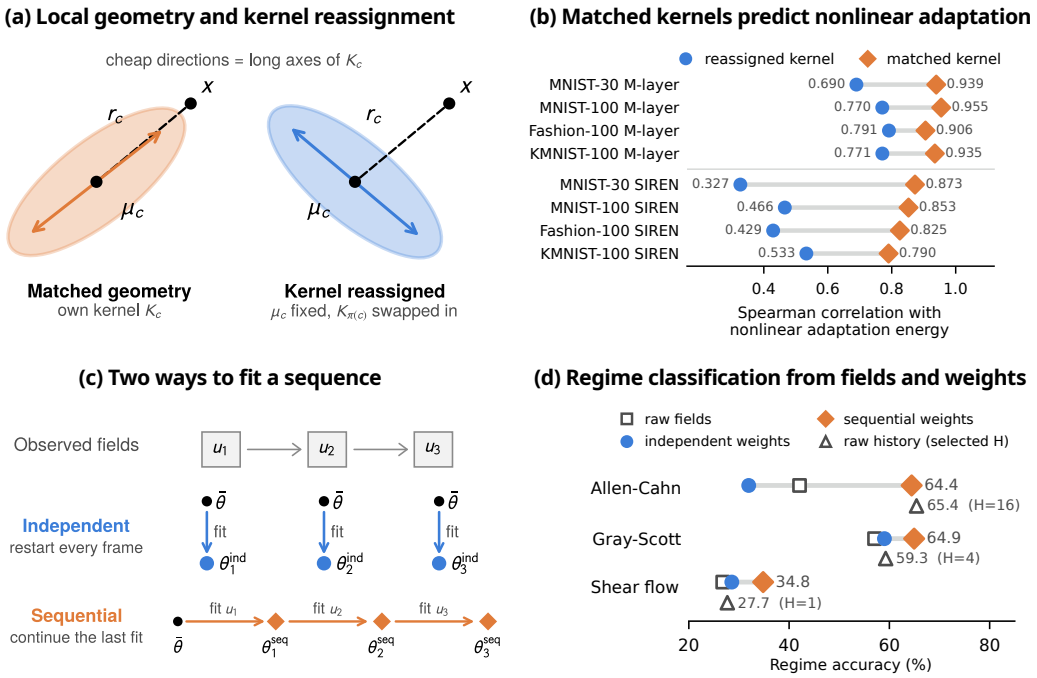


Figure 1: **Adaptation geometry and sequential fitting carry information beyond reconstruction. (a)** A class prior is its reconstruction μ_c and tangent kernel K_c that sets which output changes are cheap; reassignment keeps μ_c and swaps K_c. **(b)** In all settings, the matched kernel predicts nonlinear adaptation far better than a reassigned one (Spearman correlation with nonlinear adaptation energy). **(c)** Independent encoding restarts every frame from the shared prior $\bar{\theta}$; sequential encoding starts each fit from the previous frame's weights. **(d)** With the same class-conditioned readout, sequential weights classify the regime better than raw fields or independent weights on Allen–Cahn, Gray–Scott, and shear-flow (seed means); raw frames with explicit delays match them on Allen–Cahn but not on Gray–Scott. The shear flow delay search selects current frames alone (Table 2).

Past observations can therefore carry information about unresolved state that the current observation alone does not. We ask whether sequential neural-field fitting can retain some of this information in its weights. Continuing each frame's fit from the previous frame does so: we establish memory directly with wave histories that end at identical observations, and show downstream value in physical-regime classification (Sections 4.2 and 4.3).

The two results are connected: the same tangent spectrum that determines which changes are easy to make also determines, locally, which changes are quickly overwritten. Sequential fitting therefore turns adaptation geometry into a set of memory timescales.

Finally, a controlled geometric experiment shows that the sampled reconstruction does not uniquely determine adaptation geometry: a network can follow a closed loop in output space and return to the same reconstruction with a different geometry (Section 4.4).

2 Related Work

Neural fields, learned initializations, and local geometry. Neural fields represent signals with coordinate networks (Sitzmann et al., 2020b), and prior work learns from fitted functions or their weights (Dupont et al., 2022; De Luigi et al., 2023; Navon et al., 2023; Zhou et al., 2023). Meta-learning has also been used to learn neural-field initializations that adapt quickly to new signals (Finn et al., 2017; Nichol et al., 2018; Sitzmann et al., 2020a; Tancik et al., 2021; Papa et al., 2024). Related work uses local network sensitivity as a representation, as in Task2Vec (Achille et al., 2019), while tangent distance, NTKs, and linearized networks characterize local function

changes under parameter updates (Simard et al., 1993; Jacot et al., 2018; Lee et al., 2019; Maddox et al., 2021; Zhou et al., 2021). Maddox et al. (2021) use Jacobians of trained finite networks for analytic adaptation. Kernel Reboot (Mallak et al., 2026) combines tangent kernels with learned task priors for neural-field reconstruction and adaptation. Here we test empirically whether prior-specific tangent kernels predict rankings and choices of finite-budget nonlinear adaptation.

Sequential neural representations and memory. Warm-starting neural fields across time is not itself new: NIRVANA (Maiya et al., 2023) initializes successive video representations from previous fitted weights. Woo et al. (2025) target catastrophic forgetting in continual neural fields, while PDE methods evolve neural representations through time (Du & Zaki, 2021; Bruna et al., 2024; Chen et al., 2023). Our contribution is not sequential warm-starting itself, but the analysis of what ordinary reconstruction-based fitting leaves in its weights and whether that history is useful.

Memory in partially observed dynamics. Projecting a Markovian system onto an incomplete set of observables yields exact reduced dynamics with a Markovian term, a memory integral over the past of the observables, and a term driven by the unresolved initial state. This is the Mori–Zwanzig formalism (Mori, 1965; Zwanzig, 1961), later recast as optimal prediction (Chorin et al., 2000) and connected to learned recurrent closures (Ma et al., 2018). Buitrago Ruiz et al. (2025) use this perspective to motivate memory for learned PDE surrogates and show that explicit memory helps precisely when the state is observed at low resolution or with noise. Optimization history is also known to affect network parameters and retention (Ash & Adams, 2020; Kirkpatrick et al., 2017). We connect these views by asking whether sequential fitting itself provides a memory-bearing representation, and whether the same local adaptation geometry that controls what is easy to change also controls what is retained.

3 METHODS

Let $f_\theta(q)$ map a coordinate q to an image intensity or physical-field value. We write $F(\theta) \in \mathbb{R}^m$ to refer to its flattened outputs. Fitting $F(\theta)$ to an observation x stores a representation in θ; evaluating the network decodes its reconstruction. A *prior* here refers to a learned initialization for such a network. We learn it with Reptile (Nichol et al., 2018), which iteratively fits a network to random samples from the data and moves the initialization toward the resulting weights. For image experiments we learn a prior for each class. Sequential experiments use one shared prior across labels.

3.1 MEASURING WHICH OUTPUT CHANGES ARE CHEAP

We learn one initialization/prior θ_c for each image class, then adapt each network to reconstruct a new image and measure how much it must change to fit that image well. We compare this full adaptation with a local tangent-kernel approximation, and swap tangent kernels between classes to test whether the network's specific local geometry matters. For a class prior θ_c, define the adaptation energy of a query x as

$$E_c(x) = \min_\delta \left[\frac{1}{2\sigma^2} \|x - F(\theta_c + \delta)\|_2^2 + \frac{\lambda}{2} \|\delta\|_2^2 \right]. \tag{1}$$

Reported nonlinear adaptation scores are the lowest objective values observed during the prescribed finite-budget Adam fitting procedure; Eq 1 defines the ideal optimization objective. The first term rewards reconstruction; the second penalizes movement from the prior. Here $\sigma^2 > 0$ scales reconstruction error and $\lambda > 0$ sets the movement penalty. "Cheap" means a low value of this objective, not necessarily fewer optimizer steps. Selecting the class under which an observation has lowest energy gives an adaptation-based classifier, somewhat analogous to compression-based classifiers which check under which encoding a new observation has the smallest representation. However, our purpose is not to build a competitive classifier but rather to use this task to study the network's response to a new target.

Let $\mu_c = F(\theta_c)$, $r_c = x - \mu_c$, and $J_c = \partial F / \partial \theta|_{\theta_c}$. The Jacobian J_c maps a small weight change to its predicted output change. Its *tangent kernel*, $K_c = J_c J_c^\top$, describes the output directions that

these weight changes make accessible. Replacing the network by $F(\theta_c + \delta) \approx \mu_c + J_c\delta$ gives the exact minimum of the linearized objective:

$$E_c^{\text{lin}}(x) = \tfrac{1}{2} r_c^\top C_c^{-1} r_c, \qquad C_c = \sigma^2 I + \lambda^{-1} K_c. \tag{2}$$

This is a *Mahalanobis distance*. Under this metric, different output directions are weighted differently. Directions with large kernel eigenvalues are cheaper to change along. Thus the local description is (μ_c, K_c), the initial reconstruction together with its accessible changes, rather than μ_c alone. Eq. 2 uses the actual finite network, not an infinite-width approximation. It is exact for the linearized optimization problem; agreement with finite-step nonlinear fitting must be measured. Appendix A gives the derivation, loss normalization, and dependence on the parameter-movement metric.

3.2 Encoding a sequence by continuing the previous fit

We encode a sequence of physical fields by repeatedly fitting the same network, using the weights fitted to one frame as the starting point for the next.

This construction is motivated by settings where a single observed frame does not determine the relevant dynamics. Sequential fitting gives the representation a simple way to carry information from earlier observations forward, without adding a separately trained recurrent model.

We compare these sequential weight trajectories with fitting every frame independently and with using the raw field sequence directly, then test how much information about the underlying dynamics each representation retains. For observed frames $u_1, \ldots, u_T$, let $\mathrm{Adapt}(\theta, u)$ run a fixed fitting procedure initialized at θ. Starting from a shared prior $\bar{\theta}$, we compare

$$\begin{aligned} \text{Independent:} \quad & \theta_t^{\text{ind}} = \mathrm{Adapt}(\bar{\theta}, u_t), \\ \text{Sequential:} \quad & \theta_t^{\text{seq}} = \mathrm{Adapt}(\theta_{t-1}^{\text{seq}}, u_t), \qquad \theta_0^{\text{seq}} = \bar{\theta}. \end{aligned} \tag{3}$$

Independent fitting starts over for every frame. Sequential fitting carries the previous weights forward. Within a comparison, both use the same architecture and fitting budget. The optimizer state is reset at every frame, so information passes between fits through the weights and not through retained Adam momentum.

Reading out physical regimes. We apply the same classification procedure to raw-frame sequences, independent-weight sequences, and sequential-weight sequences. Features are standardized and compressed by principal component analysis (PCA), fitted only on support data. In the resulting coordinates z_t, each class learns a ridge-regularized linear next-step model,

$$z_{t+1} - z_t \approx A_c z_t + b_c, \qquad S_c = \frac{1}{T-1} \sum_{t=1}^{T-1} \|z_{t+1} - z_t - A_c z_t - b_c\|_2^2. \tag{4}$$

The predicted class is the one which minimizes S_c. PCA rank and ridge strength are selected on validation data using the same protocol for each representation. All three methods receive the observation sequence. Separately, the wave-memory experiment uses only the final weights θ_T as the probe input.

3.3 Local geometry sets memory timescales

The same local geometry that determines which output changes are easy to make also determines how quickly earlier parameter changes are overwritten. Under gradient descent on a frozen Jacobian with half-squared error, a perturbation along tangent mode i, with kernel eigenvalue κ_i, retains the fraction

$$\rho_i(s) = (1 - \eta\kappa_i)^s \tag{5}$$

after s steps. For $0 \le \eta\kappa_i \le 1$, larger-κ_i modes decay faster. Large-κ_i modes are therefore easy to change but quickly overwritten, while small-κ_i modes change more slowly and persist for longer. We test this prediction at sequentially fitted networks in Section 4.4. This is a statement about the controlled linearized update, not the practical Adam encoder.

A closed-loop test of adaptation geometry To test whether the current sampled output uniquely determines local adaptation geometry, we make a network follow a closed loop of target fields. At each point we use the minimum-norm parameter velocity that realizes the requested output velocity,

$$\dot{\theta} = J_\theta^+ \dot{y}. \tag{6}$$

The final target is identical to the first. We therefore measure whether the network's parameters and tangent kernel also return to their starting values. Full numerical checks, small-loop curvature diagnostics, and controls are given in Appendix G.

4 Experiments and results

We first test whether local geometry predicts adaptation and then whether sequential fitting retains useful history. We next connect these results through local retention dynamics. Finally, we include a controlled existence test showing that identical sampled outputs need not imply identical adaptation geometry. Appendices B–G contain the full protocols and numerical controls.

4.1 Local geometry predicts nonlinear adaptation

Setup. We learn class priors for MNIST, Fashion-MNIST, and KMNIST (LeCun et al., 1998; Xiao et al., 2017; Clanuwat et al., 2018). The eight settings combine dataset, number of labeled support images, and architecture choices. We use a sinusoidal network, SIREN (Sitzmann et al., 2020b) and a structured M-layer architecture (Fischbacher et al., 2020). Justification for these architectures can be found in Appendix B.2. Each uses three data seeds and held-out queries. Tuning is done on MNIST-30 (the low-data MNIST setting with 30 labeled support images in total, balanced as three examples per digit class) and frozen elsewhere. We compare a prior's own kernel with a kernel shared across classes and with kernels reassigned from other classes. Reassignment keeps each initial reconstruction μ_c fixed *in the scoring formula*; it is not an edit of the network's actual weights.

Results. Table 1 separates two prediction measures: Spearman correlation asks whether candidate scores have similar rankings to those produced by full adaptation; decision agreement asks whether they choose the same prior. On MNIST-100, the M-layer's own kernel gives 0.955 correlation and 88.3% decision agreement, compared with 0.886 and 75.7% for shared geometry. Across 27 independently learned prior sets per architecture, the own-kernel advantage over shared geometry persists on MNIST-100 (Appendix B.4).

The own-kernel improvement is also stable under resampling. On MNIST-100 with the M-layer, the own-versus-shared gain is $+0.069$ in Spearman correlation (95% CI $[0.061, 0.078]$) and $+12.7$ percentage points in decision agreement (95% CI $[8.7, 16.7]$). Full query, fold, and independent-prior uncertainty analyses are reported in Appendix B.4.

The most consistent result is the reassignment control. Giving a prior another class's kernel reduces both measures in all settings. On MNIST-100, correlation falls from 0.955 to 0.770 for the M-layer and from 0.853 to 0.466 for SIREN. Equalizing kernel traces (their total scale) preserves the gap in every setting. Thus the pairing between a prior's reconstruction and its local geometry matters, beyond just the total kernel scale.

Kernels also transfer across independent fits: same-class donors predict nonlinear adaptation rankings better than different-class donors for both architectures, including after trace normalization, while own kernels remain strongest (Appendix B.5).

Thus, *local geometry helps predict which adaptations a fitted network favors.* The connection to the sequence experiments is not that the image-class kernels transfer to physical systems; it is that the same local object, the tangent spectrum of the currently fitted network, appears in a local model of its response to the next fitting update. We now ask what repeated applications of those updates retain over time.

Table 1: **The correct local geometry predicts nonlinear adaptation.** Correlation measures similarity in score rankings; agreement measures selection of the same prior. Shared uses the mean kernel; Match uses each prior's own kernel; Shuffle reassigns kernels while fixing initial reconstructions (100 derangements). ΔAdapt. is nonlinear-minus-static classification accuracy in percentage points, a separate measure of utility. Entries are pooled held-out metrics rather than seed means; paired uncertainty for the method differences is reported in Appendix B.4.

		Spearman ρ			Decision agreement (%)		
Setting	ΔAdapt.	Shared	Match	Shuffle	Shared	Match	Shuffle
MNIST-30 M-layer	+12.2	0.829	**0.939**	0.690	73.3	**84.4**	61.9
MNIST-100 M-layer	+5.3	0.886	**0.955**	0.770	75.7	**88.3**	58.9
Fashion-100 M-layer	+2.0	**0.918**	0.906	0.791	77.0	**89.3**	61.9
KMNIST-100 M-layer	+0.0	0.884	**0.935**	0.771	65.0	**79.7**	47.7
MNIST-30 SIREN	+4.4	0.618	**0.873**	0.327	38.9	**57.8**	11.0
MNIST-100 SIREN	−15.3	0.733	**0.853**	0.466	33.0	**58.3**	6.6
Fashion-100 SIREN	−19.0	0.633	**0.825**	0.429	29.0	**66.7**	5.3
KMNIST-100 SIREN	−22.3	0.686	**0.790**	0.533	12.3	**48.3**	6.0

4.2 Different histories remain distinguishable at the same observation

To test memory directly, we construct pairs of wave trajectories with different histories but exactly the same final observed displacement. Their final velocities are opposite. A deterministic classifier given only that final observation must make the same prediction for both members of a pair. With both opposite labels present, it is correct on exactly one: accuracy is exactly 50%. Independent fitting obeys the same bound when identical observations use the same initialization and fitting.

Setup. The experiment uses 3×3 data and encoder seeds and 384 unique physical pairs. The 2,241-parameter SIREN encoder is selected using reconstruction alone and frozen before hidden-velocity labels are accessed. We verify exact observation equality and bit-identical independent encodings (Appendix E).

Results. Linear probes on sequential endpoint weights reach 68.6% mean sign accuracy, with a minimum of 64.8% across cells. The gain over independent weights is positive in all nine settings, with paired bootstrap 95% interval $[16.5, 20.6]$ percentage points. Velocity regression gives $R^2 = 0.22$, versus zero for the current-observation controls. The field decoded from the sequential endpoint reaches 55.6%, and the endpoint reconstruction residual reaches 56.4%. Thus the sequential weights make historical information more accessible to this probe, but the experiment does not establish that this information is entirely absent from the reconstruction (Figure 2). The endpoint recovery is strongest with the full 16-frame history, although it does not increase monotonically with history length, and it is essentially unchanged across per-frame budgets of 25 to 500 steps (Appendix E.3).

4.3 Sequential weights improve physical-regime classification

We encode a physical trajectory by repeatedly fitting the same small neural field, using the weights fitted to one frame as the starting point for the next. We then ask whether the resulting weight sequence makes the underlying dynamics easier to identify than either the raw field sequence or a sequence of networks fitted independently to each frame.

If history contains information about unresolved state or unknown dynamics, then a representation with memory may have something useful to retain. Our first question is therefore whether sequential fitting turns the fitting path into a useful state variable: does carrying the previous network forward make the observed dynamics easier to distinguish than representing each frame independently? The source of historical information differs across these tasks: the wave experiment omits velocity, and the shear-flow and Gray–Scott fields are spatially reduced, so both are partial observations in the Mori–Zwanzig sense, whereas Allen–Cahn observes u on the simulated grid and treats the reaction rate a as unknown, which makes it a parameter-identification task rather than a reduced-state one.

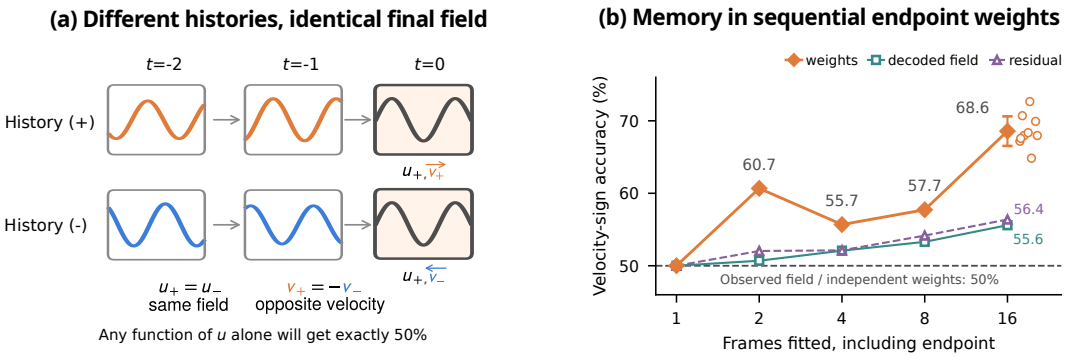


Figure 2: **Different histories stay distinguishable at the same observed endpoint. (a)** Paired wave histories differ in their earlier frames but end at exactly the same displacement field with opposite hidden velocities. **(b)** Velocity-sign accuracy of a linear probe on sequential endpoint weights as a function of how many frames were fitted before the endpoint; the observed field and independently fitted weights sit at chance by construction, and the field decoded from the sequential weights and its residual reach 55.6% and 56.4%. Dots at 16 frames are the nine seed cells; the whisker is the paired-bootstrap 95% interval.

Table 2: **Sequential fitting makes weight sequences more useful for classifying dynamics.** Accuracy (%) as seed mean / score ensemble; an ensemble standardizes each query's class-score vector within a run and averages across runs before deciding. Raw + delays selects among stacks of up to 16 raw frames in total: the current frame and at most 15 preceding frames. It reports means over three data seeds for Allen–Cahn and Gray–Scott, and one deterministic run for shear flow, where validation selects $H = 1$ (no added history). The delay baseline uses the primary evaluation, whose Gray–Scott raw comparator is 56.3 rather than the diagnostic 57.0 shown here. The final column is the sequential seed mean minus the best other mean or deterministic result; differences are computed before rounding. Chance is 25% for Allen–Cahn and shear flow and 16.7% for Gray–Scott. Allen–Cahn and Gray–Scott cross three data seeds with three encoder seeds; shear flow uses one support split and three encoder seeds.

System	Raw fields	Independent weights	Sequential weights	Raw + delays	Seq. minus best
Allen–Cahn	42.1 / 48.8	31.9 / 37.5	**64.4 / 66.2**	65.4	−1.0
Gray–Scott	57.0 / 53.3	59.0 / 55.6	**64.9 / 65.6**	59.3	+5.7
Shear flow (45 support/class)	27.7	28.6 / 33.0	**34.8 / 44.6**	27.7 ($H = 1$)	+6.2

Setup. We use three physical systems with different sources of temporal information. Allen–Cahn is a scalar reaction–diffusion system whose four classes are different reaction rates (Allen & Cahn, 1979). Gray–Scott is a two-species reaction–diffusion system with six pattern regimes (Pearson, 1993; Ohana et al., 2024). Shear flow is a fluid simulation from The Well with four Reynolds-number classes (Ohana et al., 2024). We apply the readout of Equation 4 to each system. The Gray–Scott and shear-flow fields are spatially reduced and therefore partially observed in the Mori–Zwanzig sense, whereas Allen–Cahn observes u on the simulated grid and treats the reaction rate a as unknown, making it a parameter-identification task rather than a reduced-state one. Encoders have 337 to 388 parameters. Allen–Cahn and Gray–Scott cross three data seeds with three encoder seeds.

Results. Sequential weights outperform both raw fields and independent weights on all three systems in Table 2. The stronger raw-history baseline separates the cases. On Gray–Scott, sequential weights reach 64.9% seed-mean accuracy versus 59.3% for validation-selected raw delays; on shear flow they reach 34.8% versus 27.7%, and validation selects no added raw history ($H = 1$). On Allen–Cahn, by contrast, explicit raw delays match the sequential result (65.4% versus 64.4%). Thus the clearest sequential-weight advantage is on the reduced-field Gray–Scott and shear-flow tasks, while Allen–Cahn provides a useful boundary case. Full seed-level and per-class results are in Appendix D.4.

Table 3: **More fitting improves reconstruction but eventually hurts the history-bearing representation.** Validation results for sequential fitting. Reconstruction error decreases monotonically with the number of fitting steps, while regime-classification accuracy peaks at an intermediate budget.

System	Steps/frame	Seq. MSE	Seq. accuracy
Allen–Cahn	50	0.059	64.6
	200	0.043	**69.4**
	1000	**0.035**	60.1
Gray–Scott	50	0.679	**69.2**
	100	0.599	68.8
	1000	**0.395**	64.1

What the controls establish. On shear flow, applying the same readout to fields decoded from sequential weights gives 26.8%, compared with 44.6% for the weights themselves. Removing the extra penalty for moving from the preceding fit does not remove the gain (45.5% with $\beta = 0$). Additional intermediate checkpoints and an objective encouraging different fitting routes to produce the same representation also fail to improve performance. The validation-selected raw-delay baseline closes the sequential advantage on Allen–Cahn; the advantage remains on Gray–Scott and shear flow, where the latter search selects no added history (Table 2, Appendix D.3).

4.4 The same geometry controls adaptation and forgetting

The correct Jacobian matters in a diagnostic update model, on one system. We approximate each transition by a regularized tangent update toward the next frame, then substitute a Jacobian from the wrong transition. On Gray–Scott, matched updates yield 46.5% mean classification accuracy versus 30.1% for mismatched updates, positive in all nine cells. The same intervention fails on shear flow, where matched and mismatched updates are both near chance (ensemble accuracies 26.8% versus 28.6%), so the diagnostic supports a system-dependent role for Jacobian identity rather than a general mechanism (Appendix F.1).

Reachability and forgetting share one spectrum. The tangent kernel first appeared in Section 4.1 as a measure of *reachability*: directions with larger kernel eigenvalues can be produced by smaller parameter changes. The same eigenvalues also determine how quickly information in those directions is overwritten by subsequent fitting. To see this, consider gradient descent on a frozen local linearization. A perturbation along a mode with curvature κ_i retains the fraction $\rho_i(s) = (1 - \eta\kappa_i)^s$ after s steps of learning rate η. Thus the spectrum induces a set of local memory timescales. Large-κ_i modes are easy to write but are overwritten quickly; small-κ_i modes are harder to change but persist for longer. In this local model, *adaptability and memory are therefore controlled by the same geometry*.

This prediction is quantitatively accurate under the controlled updates we test. At saved sequential endpoints, predicted and measured mode-wise retention have median rank correlations of 0.986 on Allen–Cahn and 0.945 on Gray–Scott, with log-retention slopes near the predicted value of one. After 16 steps, the slowest quartile of modes retains $0.998/0.999$ of its perturbation, while the fastest quartile retains only $0.259/0.579$ (Appendix F). This establishes the predicted local retention timescales, but not that the slow instantaneous modes carry the useful classification memory: output-matched selective erasure does not consistently damage performance more when slow modes are removed (Appendix F.4).

The practical encoder shows the corresponding budget trade-off. Increasing the fitting budget monotonically improves reconstruction, but classification peaks at an intermediate number of updates and then declines (Table 3). This is what we would expect if finite-budget fitting acts as a mode-dependent temporal filter: each new frame is written strongly into fast tangent modes while information in those modes is rapidly overwritten, whereas slower modes preserve effects of earlier frames for longer. The wave experiment does not show this washout over its tested range of 25 to 500 steps per frame (Appendix E.3); its probe is supervised, its encoder is about six times larger, and the

velocity signal may occupy slow modes, so the budget trade-off is established for the regime tasks and open for the wave task.

Equation 5 is exact for the controlled linearized update rather than the practical Adam encoder. Differentiating through the actual Adam fits passes one-frame numerical checks but becomes unstable over an eight-frame horizon, so we do not perform the corresponding causal erasure test; which directions carry useful finite-horizon memory therefore remains open (Appendix F.5).

Controlled loop at fixed sampled outputs. In a controlled transport experiment on a complete 12×12 grid, the median relative output error after a closed loop is 3×10^{-14} across nine cells at the two smallest loop scales, while the parameters and tangent kernel remain changed. This is an existence result at fixed sampled outputs; ordinary SGD and Adam do not reproduce the same transport signatures (Appendix G).

5 Discussion

What should we evaluate in a fitted representation? Our main results point to two properties beyond reconstruction: how a fitted network responds to further updates, and what repeated fitting retains from earlier observations. The tangent spectrum connects these two locally by determining both reachability and forgetting timescales. The controlled-loop experiment adds a separate geometric observation: even fixing the sampled reconstruction does not, in general, fix this local adaptation geometry.

6 Limitations

The main unresolved question is how completely the local tangent picture explains the memory produced by practical sequential fitting. The retention rule is exact for a frozen linearization under controlled gradient descent, while the practical encoder uses nonlinear Adam updates. Our results connect tangent geometry to memory timescales, but do not establish that the instantaneous tangent modes are the causal carrier of the full classification advantage. Consistent with this distinction, selective erasure does not identify the slow instantaneous modes as the useful-memory carrier, and a finite-horizon analysis through the actual Adam fits fails numerical prerequisites before its causal erasure test (Appendices F.4 and F.5).

The sequence experiments also measure usefulness under one deliberately matched class-conditioned dynamics readout. They show that sequential weights expose useful history to this readout, not that they outperform every possible classifier on the raw sequence. Likewise, in the endpoint-collision experiment the final *observations* are identical, but the sequentially decoded fields are not perfectly identical; the result therefore establishes retained history rather than completely output-independent hidden state.

Finally, the tangent geometry is defined jointly by the network, loss, and parameter-movement metric, so its numerical form is parameterization-dependent. Broader architectures and modalities will be needed to determine how generally the observed geometry–memory relationship holds.

7 Conclusion

Reconstruction is only one aspect of a neural-field representation. A fitted network also has a measurable local response to change, and repeated fitting can leave useful history in its weights. The practical implication is to evaluate these properties rather than assume that better reconstruction necessarily gives a better representation.

AI use statement

In this work, we used generative AI tools to assist with programming, experimental design and execution, analysis of results, literature search, and drafting and editing the manuscript. We did not use generative AI tools to generate synthetic datasets or to prove mathematical claims. All AI-assisted code was reviewed and tested by the authors, AI-suggested references were checked

against the original publications, and all derivations in the appendices were verified by hand. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

This work uses public image benchmarks, public physics simulations from The Well, and synthetic physical fields. It involves no human-subject experiments or personal data. We report unsuccessful as well as successful experiments, preserve held-out evaluation, and distinguish negative results from tests that could not meet numerical prerequisites.

Reproducibility statement

The appendices document the data splits, architectures, fitting procedures, hyperparameter selection, readouts, uncertainty analyses, and numerical controls used for the reported experiments. They also distinguish held-out and validation evaluations and record failed or numerically ineligible tests. An anonymized source-code repository accompanies the submission, containing the implementations, frozen numerical configurations, documented protocols, and instructions for obtaining the data and reproducing the reported experiments. The supplied commands generate the experiment outputs.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6429–6438, 2019. doi: 10.1109/ICCV.2019.00653.

Samuel M. Allen and John W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Joan Bruna, Benjamin Peherstorfer, and Eric Vanden-Eijnden. Neural galerkin schemes with active learning for high-dimensional evolution equations. *Journal of Computational Physics*, 496: 112588, 2024. doi: 10.1016/j.jcp.2023.112588.

Ricardo Buitrago Ruiz, Tanya Marwah, Albert Gu, and Andrej Risteski. On the benefits of memory for modeling time-dependent PDEs. In *International Conference on Learning Representations*, 2025.

Peter Yichen Chen, Jinxu Xiang, Dong Heon Cho, Yue Chang, G. A. Pershing, Henrique Teles Maia, Maurizio M. Chiaramonte, Kevin T. Carlberg, and Eitan Grinspun. CROM: Continuous reduced-order modeling of PDEs using implicit neural representations. In *International Conference on Learning Representations*, 2023.

Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman. Optimal prediction and the Mori-Zwanzig representation of irreversible processes. *Proceedings of the National Academy of Sciences*, 97(7): 2968–2973, 2000.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical Japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Luca De Luigi, Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Deep learning on implicit neural representations of shapes. In *International Conference on Learning Representations*, 2023.

Yifan Du and Tamer A. Zaki. Evolutional deep neural network. *Physical Review E*, 104:045303, 2021. doi: 10.1103/PhysRevE.104.045303.

Emilien Dupont, Hyunjik Kim, S. M. Ali Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Thomas Fischbacher, Iulia M. Comsa, Krzysztof Potempa, Moritz Firsching, Luca Versari, and Jyrki Alakuijala. Intelligent matrix exponentiation. *arXiv preprint arXiv:2008.03936*, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

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.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Chao Ma, Jianchun Wang, and Weinan E. Model reduction with memory and the machine learning of dynamical systems, 2018. URL `https://arxiv.org/abs/1808.04258`.

Wesley J. Maddox, Shuai Tang, Pablo Garcia Moreno, Andrew Gordon Wilson, and Andreas Damianou. Fast adaptation with linearized neural networks. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Shishira R. Maiya, Sharath Girish, Max Ehrlich, Hanyu Wang, Kwot Sin Lee, Patrick Poole, Tom Goldstein, and Abhinav Shrivastava. NIRVANA: Neural implicit representations of videos with adaptive networks and autoregressive patch-wise modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14378–14387, 2023. arXiv:2212.14593.

Amir Mallak, Alaa Maalouf, Lior Wolf, Daniela Rus, and Dan Rosenbaum. Kernel reboot: Breaking the boundaries of neural tangent kernels for neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(9):10940–10957, 2026. ISSN 1939-3539. doi: 10.1109/tpami.2026.3692624. URL `http://dx.doi.org/10.1109/TPAMI.2026.3692624`.

Hazime Mori. Transport, collective motion, and Brownian motion. *Progress of Theoretical Physics*, 33(3):423–455, 1965. doi: 10.1143/PTP.33.423.

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, et al. The well: A large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Samuele Papa, Riccardo Valperga, David Knigge, Miltiadis Kofinas, Phillip Lippe, Jan-Jakob Sonke, and Efstratios Gavves. How to train neural field representations: A comprehensive study and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

John E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.

Patrice Simard, Yann LeCun, and John S. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems*, volume 5, 1993.

Vincent Sitzmann, Eric R. Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. MetaSDF: Meta-learning signed distance functions. In *Advances in Neural Information Processing Systems*, volume 33, 2020a.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, 2020b.

Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2846–2855, 2021.

Seungyoon Woo, Junhyeog Yun, and Gunhee Kim. Meta-continual learning of neural fields. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 56076–56098, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/8ccc5ec30a8d46793d790e2216efd40d-Paper-Conference.pdf.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J. Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yufan Zhou, Zhenyi Wang, Jiayi Xian, Changyou Chen, and Jinhui Xu. Meta-learning with neural tangent kernels. In *International Conference on Learning Representations*, 2021.

Robert Zwanzig. Memory effects in irreversible thermodynamics. *Physical Review*, 124(4):983–992, 1961. doi: 10.1103/PhysRev.124.983.

A LOCAL ADAPTATION: DERIVATIONS AND CONTROLS

A.1 THE OBJECTIVE AND ITS IMPLEMENTED NORMALIZATION

The adaptation objective combines reconstruction error with the cost of moving away from a learned initialization. The more general form allows different parameter directions to have different movement costs:

$$\mathcal{E}_c(x) = \min_{\delta} \left[\frac{1}{2\sigma^2} \|x - F(\theta_c + \delta)\|_2^2 + \frac{1}{2} \delta^\top \Lambda_c \delta \right], \qquad \Lambda_c \succ 0. \tag{7}$$

The positive-definite matrix Λ_c specifies the movement metric. The image experiments use the isotropic special case

$$E_c(x) = \min_{\phi} \left[\mathrm{MSE}(x, F(\phi)) + \beta \|\phi - \theta_c\|_2^2 \right], \qquad \beta = 10^{-3}. \tag{8}$$

When the mean squared error averages over m scalar targets, the two expressions agree for $\sigma^2 = m/2$ and $\Lambda_c = \lambda I$ with $\lambda = 2\beta$. For the five-field M-layer, $m = 3920$, $\sigma^2 = 1960$, and $\lambda = 0.002$. For the intensity-only SIREN, $m = 784$, with the same conversion. The implemented nonlinear score is the minimum objective value encountered during finite-step fitting, not a guaranteed global minimum.

A.2 SOLVING THE LINEARIZED ADAPTATION PROBLEM

Write $\mu_c = F(\theta_c)$, $r_c = x - \mu_c$, and $J_c = \partial F / \partial \theta|_{\theta_c}$. Substituting the local approximation $F(\theta_c + \delta) \approx \mu_c + J_c \delta$ gives

$$Q_c(\delta) = \frac{1}{2\sigma^2} \|r_c - J_c \delta\|_2^2 + \frac{1}{2} \delta^\top \Lambda_c \delta.$$

For this derivation only, let $H_c = \Lambda_c + \sigma^{-2} J_c^\top J_c$ and $g_c = \sigma^{-2} J_c^\top r_c$. Setting the derivative to zero gives $\delta_c^* = H_c^{-1} g_c$ and

$$\min_{\delta} Q_c(\delta) = \frac{1}{2\sigma^2} r_c^\top r_c - \frac{1}{2} g_c^\top H_c^{-1} g_c.$$

Woodbury's matrix identity rewrites this expression in output coordinates:

$$E_c^{\mathrm{lin}}(x) = \tfrac{1}{2} r_c^\top C_c^{-1} r_c, \qquad C_c = \sigma^2 I + J_c \Lambda_c^{-1} J_c^\top. \tag{9}$$

This calculation uses the Jacobian of the trained finite network. It does not assume that the network is infinitely wide or remains linear throughout nonlinear adaptation.

For $\Lambda_c = \lambda I$, diagonalize $K_c = J_c J_c^\top$ as $U_c \,\mathrm{diag}(\kappa_{c,i}) U_c^\top$. If $u_{c,i}$ is the corresponding output direction, then

$$E_c^{\mathrm{lin}}(x) = \frac{1}{2} \sum_i \frac{(u_{c,i}^\top r_c)^2}{\sigma^2 + \kappa_{c,i}/\lambda}. \tag{10}$$

This makes the direction-dependent cost explicit. A residual aligned with a large kernel eigenvalue contributes less energy than an equally large residual along a weak direction. The prior output is the center of the local model; its kernel determines the shape of the surrounding equal-cost ellipsoids.

A.3 RELATION TO THE NONLINEAR OBJECTIVE NEAR THE PRIOR OUTPUT

The tangent energy is also the quadratic Taylor term of the nonlinear adaptation energy at the prior output $x = \mu_c$. Assume F is twice continuously differentiable near θ_c. At $r = x - \mu_c = 0$, the minimizing displacement is zero, and the Hessian with respect to displacement is $H_c \succ 0$. The implicit-function theorem gives a locally smooth minimizing displacement as a function of r.

Differentiating the objective after minimizing over displacement yields

$$\nabla_r \mathcal{E}_c(\mu_c) = 0, \qquad \nabla_r^2 \mathcal{E}_c(\mu_c) = \sigma^{-2} I - \sigma^{-4} J_c H_c^{-1} J_c^\top = C_c^{-1}.$$

Hence

$$\mathcal{E}_c(\mu_c + r) = \tfrac{1}{2} r^\top C_c^{-1} r + o(\|r\|_2^2).$$

This is a local statement about the objective. It does not guarantee accurate predictions for large adaptations or for the final output of a finite-step optimizer.

A.4 SHARED, REASSIGNED, AND SCALE-NORMALIZED KERNELS

The static score uses only the residual from the prior output, $\|x-\mu_c\|_2^2$. The shared-kernel score uses $\bar{K}$, the mean kernel across classes, in every candidate's distance. This allows direction-dependent adaptation, but removes candidate-specific kernel identity.

For reassignment, choose a permutation π in which no class keeps its own kernel. Such a permutation is a *derangement*. We replace C_c by

$$C_{c,\pi} = \sigma^2 I + \lambda^{-1} K_{\pi(c)},$$

while keeping the prior output μ_c fixed. Each reported shuffled value averages 100 derangements. This is an intervention on the scoring rule, not a construction of a network with the reassigned kernel.

To test whether total kernel scale explains the effect, repeat the comparison after setting

$$\widetilde{K}_c = K_c \frac{\bar{t}}{\mathrm{tr}(K_c)}, \qquad \bar{t} = \mathrm{mean}_c \,\mathrm{tr}(K_c).$$

Every kernel then has the same trace, while retaining its eigenvalue ratios and eigendirections. Persistence of the gap shows that total scale alone does not explain the matched-kernel advantage.

A.5 DEPENDENCE ON PARAMETER COORDINATES

The size of a weight change depends on the parameter coordinates used to describe it. For a locally invertible change of coordinates $\delta' = B\delta$, the Jacobian transforms as $J' = JB^{-1}$. To preserve the same movement cost, the metric must transform as $\Lambda' = B^{-\top}\Lambda B^{-1}$. The combination

$$J'\Lambda'^{-1}J'^{\top} = J\Lambda^{-1}J^{\top}$$

is then unchanged. The geometry is therefore defined by the network, loss, and movement metric together, rather than by the network function alone (Amari, 1998). Fixing a Euclidean penalty in different coordinates need not describe the same adaptation problem.

B IMAGE EXPERIMENTS: PROTOCOLS AND EXTENDED RESULTS

B.1 DATA SPLITS AND HYPERPARAMETER SELECTION

MNIST support sets contain 20, 30, or 100 balanced labels, or 2–10 examples per class. Data-selection seeds are 0–2. The 20-, 30-, and 100-label studies use respectively 2, 3, and 10 leave-one-per-class folds per seed. A query is excluded from the support set used to fit its own class prior.

A fixed 5,000-image MNIST test split is used only for the frozen 20- and 30-label evaluations, after the score and movement penalty β have been fixed. It is not loaded for score selection, tangent analysis, kernel reassignment, or bootstrap analysis. Fashion-MNIST and KMNIST use 100 balanced labels, three seeds, and 10 folds per seed, giving 300 cross-validation queries per dataset.

The movement penalty $\beta = 0.001$ was selected in a separate 20-label MNIST study. The M-layer tangent scorer and protocol selected on MNIST-30 are reused on MNIST-100 and both transfer datasets. For SIREN, a 36-configuration grid on MNIST-30 varies width over $\{32, 64\}$, frequency ω over $\{10, 30\}$, and Reptile and query learning rates over $\{5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$. It selects width 32, $\omega = 10$, and both learning rates 5×10^{-4}. This configuration is frozen for the other three SIREN settings.

The M-layer uses three structured branches to decode five target fields derived from intensity, Sobel, and distance-transform channels. Its Jacobians are evaluated in exact block form. The SIREN is a conventional two-dimensional sinusoidal coordinate network with scalar output. All ten priors within a fold start from the same base parameter vector.

At inference, each query is fitted independently from every frozen class prior. Every candidate receives fresh optimizer state, the same penalty, and at most 2,000 steps with early-stopping patience 200. The score is the lowest observed value of Equation 8; the adapted copy is discarded. No separate network is trained to classify the adapted image weights.

Table 4: **Architecture and fitting settings for the image experiments.**

Component	Five-field M-layer	Intensity-only SIREN
Outputs	$5 \times 28 \times 28 = 3920$	$28 \times 28 = 784$
Adapted parameters	913	2,241
Prior initialization	Common within each fold	Common within each fold
Reptile inner steps	8 Adam steps	8 steps
Reptile inner learning rate	0.04	5×10^{-4}
Reptile outer steps	1,000	1,000
Outer interpolation	0.1, constant	0.1, constant
Query learning rate	0.04	5×10^{-4}
Regularization	$\beta = 0.001$	$\beta = 0.001$

B.2 Choice of INR architectures

We use SIRENs and M-layers to test whether the adaptation-geometry effect persists across two substantially different parameterizations that can represent periodic structure. SIREN uses sinusoidal activations, while the M-layer's matrix-exponential nonlinearity can also represent oscillatory structure (Sitzmann et al., 2020b) Fischbacher et al. (2020).

B.3 Prediction metrics and uncertainty

Spearman correlation is computed over complete query–class score vectors. Decision agreement is the fraction of queries for which a comparison score selects the same prior as nonlinear adaptation. Both measure prediction of the adaptation procedure; neither is ground-truth classification accuracy.

Paired query bootstraps use 20,000 replicates, resampling queries within data-seed and true-class strata while retaining complete class-score vectors. Fold bootstraps use 10,000 replicates for the original M-layer MNIST-30 and MNIST-100 comparisons, and 20,000 for the SIREN comparisons and the M-layer Fashion-MNIST and KMNIST comparisons. The independent-prior experiment repeats meta-training across 27 prior sets per architecture: three data seeds, three folds, and three Reptile seeds. Query and fold intervals condition on fitted priors, while the latter experiment also varies prior fitting.

B.4 Replication and the distinction from classification accuracy

Table 5: **Replication of the own-kernel advantage over shared geometry.** $\Delta\rho$ is the correlation difference; ΔAgree is the decision-agreement difference in percentage points. Intervals are 95% confidence intervals. The last two rows refit the priors.

Setting	Replication unit	Comparisons	$\Delta\rho$	95% CI	ΔAgree	95% CI
MNIST-30 M-layer	Query bootstrap	900 pairs	+0.110	[0.093, 0.130]	+11.1	[3.3, 18.9]
	Fold bootstrap	900 pairs	+0.110	[0.090, 0.132]	+11.1	[2.2, 18.9]
MNIST-100 M-layer	Query bootstrap	3,000 pairs	+0.069	[0.061, 0.078]	+12.7	[8.7, 16.7]
	Fold bootstrap	3,000 pairs	+0.069	[0.060, 0.078]	+12.7	[8.3, 17.0]
MNIST-100 M-layer	27 prior sets	2,700 pairs	+0.069	[0.048, 0.095]	+13.7	[5.6, 23.7]
MNIST-100 SIREN	27 prior sets	2,700 pairs	+0.134	[0.096, 0.192]	+33.0	[22.6, 43.3]

For SIREN, using each prior's own kernel rather than the shared kernel improves both prediction metrics in all four settings. Correlation gains are 0.255, 0.120, 0.192, and 0.104 on MNIST-30, MNIST-100, Fashion-100, and KMNIST-100; decision-agreement gains are 18.9, 25.3, 37.7, and 36.0 percentage points. Paired query and fold intervals exclude zero for every setting. The final three are frozen evaluations.

Fashion-MNIST with the M-layer limits the shared-kernel comparison: the own kernel has slightly lower pooled correlation (0.906 versus 0.918; paired 95% interval for the difference $[-0.0227, -0.0019]$). Its decision agreement is nevertheless higher (89.3% versus 77.0%), and its own kernel outperforms reassigned kernels. The result shared by all eight settings is the benefit

Table 6: **The reassignment effect persists after equalizing total kernel scale.** Each kernel is rescaled to a common trace. Shuffle averages 100 derangements; decision agreement is in percent.

	Spearman ρ		Decision agreement	
Setting	Match	Shuffle	Match	Shuffle
MNIST-30 M-layer	0.943	0.698	82.2	62.6
MNIST-100 M-layer	0.957	0.770	89.0	58.5
Fashion-100 M-layer	0.909	0.797	87.3	62.3
KMNIST-100 M-layer	0.935	0.778	79.0	48.5
MNIST-30 SIREN	0.841	0.362	64.4	10.9
MNIST-100 SIREN	0.840	0.503	60.7	6.4
Fashion-100 SIREN	0.817	0.443	61.3	6.0
KMNIST-100 SIREN	0.772	0.554	48.0	6.4

of the correct mean–kernel pairing over reassignment, not an improvement over shared geometry on every measure.

The reassignment effect also survives equalized traces (Table 6). On MNIST-30 with the M-layer, none of the 100 unnormalized derangements matches the correct pairing in accuracy, correlation, or decision agreement. The plus-one Monte Carlo randomization value is $p = 0.0099$ for each measure.

The replicated result is prediction of nonlinear adaptation, not a superior image classifier. In the low-label comparisons, direct-image CNNs remain stronger: 82.8% and 89.9% at 20 and 30 labels, versus 72.3% and 66.4% for nonlinear adaptation.

B.5 Independent-refit kernel transfer

Both architectures show

$$\textbf{Own} > \textbf{Same-class donor} > \textbf{Shared} > \textbf{Different-class donor}$$

for within-query rank correlation and decision agreement. The pattern persists after common-trace normalization. Thus, class-associated geometry transfers across independent fits, while some useful geometry remains specific to the recipient fit. Within-query rank correlation ranks the ten candidate priors for each query separately, whereas Table 1 pools all query–class pairs; the shared kernel therefore scores lower here than in the main table.

The original-scale within-query Spearman results are:

Architecture	**Own**	**Same-class donor**	**Different-class donor**	**Shared**
M-layer	0.9117	0.8539	0.5549	0.7385
SIREN	0.7847	0.7142	-0.0802	0.0917

Same-class minus different-class Spearman is $+0.2990$ $[0.2775, 0.3214]$ for M-layer and $+0.7944$ $[0.7705, 0.8183]$ for SIREN. After trace normalization, the corresponding differences are $+0.3104$ $[0.2888, 0.3329]$ and $+0.7499$ $[0.7246, 0.7751]$, respectively. Own-minus-same-class gaps also exclude zero.

Query bootstrap intervals resample complete query vectors within data-seed/true-class strata, carrying all fits and mappings together. Intervals condition on the saved fits; recipient/donor variation is reported separately. The conclusions concern the tangent scoring surrogate, with the actual nonlinear learner held fixed.

B.6 Boundary checks for the tangent score

Predicting scores does not mean predicting final reconstructions. For nonlinear endpoint ϕ_c^*, define

$$\epsilon_{\text{lin}} = \frac{\|F(\phi_c^*) - F(\theta_c) - J_c(\phi_c^* - \theta_c)\|_2}{\|F(\phi_c^*) - F(\theta_c)\|_2 + \epsilon}.$$

On MNIST-30, means range from 0.66 to 0.88 and medians from 0.59 to 0.87 across subsets of query–class pairs. The tangent model preserves useful relative score structure without accurately reproducing the final adapted output.

The energy is not calibrated probabilistic evidence. Adding the Gaussian-evidence term $\frac{1}{2}\log\det C_c$ performs poorly under the implemented metric and noise model (10.0% classification accuracy and -0.187 correlation with nonlinear energy). The positive result therefore concerns the quadratic score, not a calibrated evidence interpretation.

C SHEAR-FLOW SPLIT INTEGRITY

Preliminary experiments exposed duplicate initial conditions in the locally packaged shear-flow data. The final support, validation, and test manifests used for the reported results were rebuilt to be disjoint in canonical initial-condition identity.

D SEQUENTIAL ENCODING: PROTOCOLS, CONTROLS, AND REPLICATION

D.1 SYSTEMS, SPLITS, AND ENCODERS

Shear flow. The task has four Reynolds classes. Support is matched across nuisance variables: each class has the same 45 combinations of Schmidt number and initial-condition identity. This gives 180 support trajectories and 675 transitions per class over 16 source frames. The four normalized channels are bilinearly downsampled to a 32×64 grid with `align_corners=False`. Validation and held-out test each contain 112 trajectories, disjoint in initial condition.

The encoder is a width-16, one-hidden-layer SIREN with $\omega = 10$ and 388 parameters. Each frame receives 50 fresh-Adam steps, starting from one shared, label-free Reptile prior for independent encoding or the preceding endpoint for sequential encoding. The initial run uses $\beta = 10^{-3}$. A fixed coordinate schedule avoids adding sampling noise to the apparent temporal motion.

Allen–Cahn. The simulated field satisfies

$$\partial_t u = 0.08\,\Delta u + a(u - u^3)$$

on a periodic 32×32 grid, with reaction-rate classes $a \in \{0.5, 1, 2, 4\}$. Each data seed has 12 support and 8 validation initial conditions per class, plus 20 fresh test initial conditions per class. Three data seeds are crossed with three encoder seeds. The width-16 SIREN has 337 parameters, uses 50 Adam steps per frame at learning rate 3×10^{-3}, and starts from a shared label-free Reptile prior. There is no movement penalty ($\beta = 0$).

Gray–Scott. The Well supplies six pattern regimes: bubbles, gliders, maze, spirals, spots, and worms. The two-channel fields are area-downsampled to 32×32. The reported experiment uses disjoint blocks within the training files: support initial conditions 0–79, with 64 used per class; validation 100–119; an unused guard block 120–144; and a held-out result block 145–159. It uses frames 500–515, a 354-parameter SIREN, 200 Reptile steps, and 100 fresh-Adam steps per frame at learning rate 3×10^{-4} with $\beta = 0$. For sequential weights, the selected dynamics rank is 64 with zero ridge penalty. Hyperparameters are selected on Gray–Scott validation initial conditions, prioritizing the worst encoder seed, and the held-out block is opened once.

D.2 MATCHED READOUT AND REPORTING CONVENTIONS

Raw frames and flattened network weights are standardized and mapped to a whitened PCA representation fitted only on support data. Each class fits a ridge-regularized linear model of successive differences, as in Equation 4. The mean squared next-step residual is the class score. PCA rank and ridge strength are selected on validation data rather than the result split.

A seed mean averages classification accuracies across fitted data/encoder cells. A score ensemble standardizes each query's class-score vector within each run, averages these vectors across runs, and then makes the class decision. Shear flow has one data split and three encoder runs; Allen–Cahn and

Gray–Scott cross three data seeds with three encoder seeds. Crossed seed cells measure different sources of variation and are not treated as independent datasets.

Gray–Scott is reported under two evaluation protocols on the same held-out block. The primary protocol reports raw fields at 56.3/57.8 and sequential weights at 64.9/65.6 (seed mean / score ensemble). Table 2 evaluates raw fields, independent weights, and sequential weights together, giving 57.0/53.3, 59.0/55.6, and 64.9/65.6, respectively. Because the sequential result is unchanged while the raw comparator differs, we keep the two protocols separate rather than pool them.

D.3 Controls on what makes the sequential weights useful

Table 7: **Shear flow results and representation controls.** Accuracy (%) is shown for the score ensemble and individual encoder seeds. Raw states and the validation-selected delay baseline are the same deterministic representation ($H = 1$). Stored parameter counts include the dynamics readout for one encoder seed, excluding PCA arrays.

Representation	Test (ensemble)	Encoder seeds	Macro-F1	Stored params
Raw states / selected delay ($H = 1$)	27.7	n/a	0.233	4,224
Independent INR weights	33.0	33.9 / 23.2 / 28.6	0.235	4,612
Sequential endpoints ($\beta = 10^{-3}$)	44.6	33.0 / 38.4 / 33.0	0.439	**676**
Sequential endpoints ($\beta = 0$)	**45.5**	33.9 / 38.4 / 33.9	0.439	676
Full adaptation path (6 checkpoints)	39.3	33.0 / 37.5 / 31.2	0.384	2,116
Decoded sequential fields	26.8	23.2 / 24.1 / 31.2	0.225	n/a

On shear flow, sequential fitting improves reconstruction MSE from about 0.758 for independent fits to 0.182. This initially leaves better reconstruction as an alternative explanation. Decoding the sequential weights and applying the same readout gives only 26.8%, compared with 44.6% for the weights. The tested decoded-field representation therefore does not reproduce the advantage; this does not prove that all possible readouts of decoded fields would fail.

Removing the movement penalty slightly improves the sequential ensemble to 45.5%. Concatenating six intermediate checkpoints per transition instead of using only frame endpoints lowers accuracy to 39.3%. The original 44.6% ensemble corresponds to 50 correct predictions out of 112.

Encouraging different fitting routes to agree. A label-free objective reduces the MSE between direct and sequential fitting routes from 0.01131 to 0.00957, but with a matched four-step SGD adapter classification reaches only 35.7%, below the 45.5% result for ordinary warm starts. Reducing this measured path dependence therefore does not improve the tested downstream task.

Explicit raw-frame delays. We test whether explicit access to observation history reproduces the sequential-weight advantage. For $H \in \{1, 2, 4, 8, 16\}$, we concatenate H raw frames, including the current frame, before support-fitted standardized PCA and apply the same class-conditioned linear dynamics scorer. Concatenation starts with the current frame and proceeds backward in time; missing early history repeats the first frame. Thus $H = 16$ includes at most 15 preceding frames. History length, PCA rank, and ridge strength are selected using the original validation procedure; $H = 1$ reproduces the current-raw decisions. On Allen–Cahn, the selected $H = 16$ delay representation reaches 65.4%, compared with 42.1% for current raw fields and 64.4% for sequential weights. On Gray–Scott, the selected $H = 4$ representation reaches 59.3%, compared with 56.3% raw and 64.9% sequential. Thus explicit raw history closes the Allen–Cahn gap under this readout, but does not reproduce the Gray–Scott result. On shear flow, we use the same normalized $32 \times 64 \times 4$ fields, support-only preprocessing, and class-conditioned dynamics scorer. We search $H \in \{1, 2, 4, 8, 16\}$, PCA ranks $\{2, 4, 8, 16, 32, 64\}$, and ridge strengths $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ on the 112 validation trajectories, selecting by validation performance with fixed tie-breaking. The selected configuration is $H = 1$, rank 32, and zero ridge. After freezing it and the PCA, test accuracy is 27.7% $(31/112)$, identical to the original raw baseline; sequential weights score 34.8% as a mean and 44.6% as an ensemble. Only the selected configuration and the $H = 1$ control are evaluated on test; longer histories are compared on validation only. The historical test set had already been used in the earlier shear-flow studies.

For Allen–Cahn and Gray–Scott, raw and delay results average three data-seed runs; the sequential result additionally crosses three encoder seeds per data seed. Shear flow has one fixed data split and one deterministic raw/delay run, compared with three existing sequential encoders. The delay search therefore does not provide three independent shear flow replications.

D.4 Replication, classwise behavior, and forecasting

The sequential-minus-raw difference is positive in every tested seed cell under the reported system-specific protocols. On shear flow, all three sequential encoder seeds beat the deterministic raw baseline. On Allen–Cahn, all nine crossed data/encoder cells are positive, with improvements ranging from 5.0 to 38.7 points. On Gray–Scott, all nine cells are positive, with a minimum improvement of 5.6 points.

Table 8: **Gray–Scott held-out stability.** Accuracy (%) and variation across encoder/data seeds for the primary held-out evaluation.

Representation	Mean	Worst seed	Seed SD	Range	Ensemble
Raw fields	56.3	54.4	1.39	3.3	57.8
Sequential weights	**64.9**	**62.2**	**1.18**	3.3	**65.6**

Gray–Scott remains uneven across classes despite its seed stability. Per-class sequential accuracies are maze 100.0%, spots 100.0%, worms 97.0%, spirals 85.9%, gliders 6.7%, and bubbles 0.0%, with seed standard deviations at most 6.4. The aggregate therefore demonstrates a reproducible gain for this task while leaving substantial classwise failure modes.

On Gray–Scott, held-out one-step R^2 is near zero or negative across ranks for all representations: raw peaks near -0.0004, while independent and sequential weights peak near $+0.0014$. Thus the Gray–Scott classification gain does not rely on an accurate low-dimensional one-step forecast.

E Wave histories with exactly identical final observations

E.1 Construction, exact bound, and encoder audit

Each pair consists of two periodic two-dimensional wave trajectories with 80 real Fourier modes, common terminal displacement coefficients and wave speed, and opposite terminal velocity coefficients. Sixteen frames span $t \in [-0.75, 0]$. At $t = 0$, observed displacement fields are exactly equal and velocities exactly opposite.

The exact 50% bound follows from the pairing. Any deterministic feature map of the terminal observation gives identical features to both members. A deterministic sign classifier must then assign the same label to both, making exactly one correct prediction per pair. The same argument applies to independent network fitting when identical observations use the same fitting procedure and random schedule. It does not apply to a sequential encoder, whose earlier observations differ.

We verify data regeneration, wave-equation residuals and finite-difference convergence, exact terminal-observation equality and opposite velocities, split integrity, exact current-observation chance, and bit-identical independent encodings

Encoder selection uses reconstruction alone. The selected width-32, two-hidden-layer SIREN has $\omega = 10$ and 2,241 parameters. Its validation frame MSE is 2.4×10^{-4}, compared with 1.0×10^{-3} for the width-16 alternative. Training uses 200 Reptile outer steps and 100 full-grid Adam steps per frame at learning rate 3×10^{-3}, with fresh optimizer state and $\beta = 0$. All nine priors are frozen before hidden-state labels or result data are opened.

Three data seeds are crossed with three encoder seeds, with 384 held-out physical pairs. Paired-bootstrap intervals resample the physical pairs after averaging encoder effects within pair; crossed data/encoder cells are not treated as independent observations.

For the main nine-cell probe experiment, standardization and PCA are fitted on support data, and probe ranks and regularization strengths are selected on validation data. Probe selection uses four

CPU threads. The decoded-field control uses float32 PCA, whose result can depend on the BLAS thread count; the supplement records this setting and the numerical software versions.

E.2 Endpoint probes and reconstruction leakage

Table 9: **Wave-memory results for each seed cell.** Sign accuracy is in percent. Current-field and independent-weight probes are exactly 50% in every row. Velocity R^2 uses sequential weights; Pair RMSE measures the difference between paired sequential reconstructions.

Data seed	Enc. seed	Sequential	Seq. decoded	Residual	Velocity R^2	Pair RMSE
14011	101	67.2	56.6	58.2	0.229	0.0062
14011	211	68.0	54.7	50.8	0.216	0.0066
14011	307	68.0	52.0	58.6	0.217	0.0065
25031	101	64.8	57.4	59.0	0.234	0.0064
25031	211	67.6	57.0	54.7	0.222	0.0066
25031	307	68.4	52.7	53.5	0.200	0.0065
36061	101	72.7	56.6	59.0	0.235	0.0062
36061	211	70.7	55.5	55.9	0.216	0.0065
36061	307	69.9	57.8	57.8	0.211	0.0065

Current-observation and independent-weight sign probes score exactly 50.0% in every cell. Sequential weights give mean accuracy 68.6%, minimum 64.8%, and seed standard deviation 2.3. The mean gain is 18.6 points, with paired bootstrap 95% interval $[16.5, 20.6]$. A ridge probe of the full terminal velocity gives $R^2 = 0.22$, while the current-observation controls give zero.

Sequential reconstructions are not exactly equal across the paired histories. Their decoded-field sign probes reach 55.6% on average, and terminal-residual probes reach 56.4%. Median pairwise reconstruction root mean squared error is 0.00645. These measurements distinguish two claims: the final observed field lacks the sign information, but the sequentially decoded field need not lack it. Above-chance prediction from weights establishes retained history, not an exclusively output-invisible memory.

E.3 Follow-up: fitting budget and history length

History length. Endpoint sign accuracy is 50.0%, 60.7%, 55.7%, 57.7%, and 68.6% for 1, 2, 4, 8, and 16 fitted observations. The 16-frame condition exceeds every shorter-history condition under paired intervals. The result supports accessible historical information at the endpoint but not monotonic accumulation with history length.

Fitting budget. From 25 to 500 Adam steps per frame, endpoint-weight accuracy remains between 67.7 and 68.6 and velocity R^2 remains between 0.215 and 0.222. Thus this range does not show useful-memory washout as fitting increases. Decoded-field and residual probes follow different budget trends, so the weight signal is not explained simply by the measured reconstruction leakage.

F Tangent diagnostics and retention tests

F.1 Matched and mismatched local updates

At a transition's current weights θ_t, let the residual to the next observation be $r_t = u_{t+1} - F(\theta_t)$. The tangent diagnostic solves

$$\delta_t^* = \arg\min_{\delta} \left[\|J_t\delta - r_t\|_2^2 + \lambda\|\delta\|_2^2 \right].$$

The mismatched control uses another transition's Jacobian, with the same targets, ridge scale, and downstream scoring. On shear flow the ridge scale is selected using support fidelity and is $\lambda = 0.1$. This changes the local update model, rather than selectively changing geometry inside the original nonlinear encoder.

Table 10: **Replacing the Jacobian changes a local update model on Gray–Scott.** Accuracy (%) over nine seed cells, using 128 Jacobian points per transition. Matched beats mismatched in all nine cells. This diagnostic run reports its own raw comparator.

Representation / update	Seed mean	Ensemble	Seed SD
Raw-state dynamics	57.0	53.3	3.8
Independent nonlinear INR	59.0	55.6	4.0
Sequential nonlinear INR	**64.9**	**65.6**	**1.2**
Matched tangent update	46.5	54.4	4.1
Mismatched tangent update	30.1	42.2	6.3

On Gray–Scott, the primary diagnostic uses all 15 transitions and 128 fixed spatial points per transition. Hyperparameters are selected on validation initial conditions 100–119 and evaluated on held-out initial conditions 145–159. Matched updates have cosine 0.159 with nonlinear displacements, compared with 0.011 for mismatched updates. Correlation of pairwise displacement geometry is 0.146 versus 0.027. Classification improves by 16.4 seed-mean points or 12.2 ensemble points, positive in all nine cells.

Sampling resolution matters. At 32 spatial points, the seed-mean gain drops to 7.4 points, with six of nine positive cells and an ensemble difference of -4.4 points. The tangent model remains an imperfect description of the nonlinear displacement even at the higher resolution.

The same intervention fails on historical shear flow. Matched ensemble accuracy is 26.8% (seeds $25.9/30.4/28.6$), versus 28.6% mismatched (seeds $24.1/33.0/32.1$) and 38.4% nonlinear. This occurs despite greater local fidelity than on Gray–Scott: matched displacement cosine and geometry correlation are $0.537/0.679$, versus $0.074/0.076$ mismatched. The nonlinear comparator differs from Table 7 because this protocol selects rank and ridge across three validation encoders. The diagnostic supports a system-dependent role for tangent identity, not a general causal explanation of nonlinear memory.

F.2 The retention rule and its conditions

Consider two runs of gradient descent on the same frozen local model and target, differing only by a small initial parameter perturbation e_0. For loss $\frac{1}{2}\|J\delta - r\|_2^2$, their difference follows

$$e_{s+1} = (I - \eta J^\top J)e_s.$$

If e_0 lies in an eigenvector of $J^\top J$ with eigenvalue κ_i, its signed amplitude after s steps is multiplied by $(1 - \eta\kappa_i)^s$. This gives Equation 5. Under $0 \le \eta\kappa_i \le 1$, larger eigenvalues correspond to faster decay without sign changes. This condition makes precise the easy-to-change, easy-to-overwrite interpretation.

For a fixed positive-definite preconditioner P, write $e = P^{1/2}\widetilde{e}$. The corresponding symmetric matrix is $P^{1/2}J^\top JP^{1/2}$, with the same nonzero eigenvalues as $JPJ^\top$. The same mode-wise rule applies in these coordinates. An alternative squared-loss normalization rescales the curvature or effective step size. A changing Jacobian, a changing preconditioner, or retained optimizer state is outside this exact frozen-model calculation.

Implementation checks validate the linear rule to maximum absolute error 1.33×10^{-15} and the preconditioned version to relative error 2.51×10^{-14}. Median nonlinear locality errors are 5.3×10^{-8}, 1.6×10^{-7}, and 5.3×10^{-7} at the small, medium, and large perturbation amplitudes tested.

At saved sequential endpoints, the controlled-update measurements agree closely with the retention rule (Table 11). Across seed combinations, median correlations range from 0.982 to 0.989 for Allen–Cahn and 0.910 to 0.964 for Gray–Scott. These controlled measurements do not make the formula exact for the practical Adam encoder.

Table 11: **Predicted and measured retention under controlled updates.** Correlation compares mode-wise retention for each transition. A log-retention slope of one is exact agreement. Slow and fast columns give retention after 16 steps.

System	Instances	Median Spearman	Median log slope	Slow ret. s=16	Fast ret. s=16
Allen–Cahn	216	0.986	1.005	0.998	0.259
Gray–Scott	324	0.945	0.992	0.999	0.579

Table 12: **More fitting improves reconstruction but not always classification.** Validation results. Raw accuracy is fixed at 41.7% for Allen–Cahn and 61.1% for Gray–Scott.

System	Steps/frame	Seq. MSE	Indep. acc.	Seq. acc.
Allen–Cahn	1	0.695	26.7	25.7
Allen–Cahn	50	0.059	37.2	64.6
Allen–Cahn	200	0.043	42.7	**69.4**
Allen–Cahn	1000	**0.035**	45.1	60.1
Gray–Scott	1	1.128	52.2	56.6
Gray–Scott	50	0.679	55.5	**69.2**
Gray–Scott	100	0.599	52.0	68.8
Gray–Scott	1000	**0.395**	57.2	64.1

F.3 Fitting budget in the practical encoder

Reconstruction improves as the fitting budget grows, but classification is not monotone. Allen–Cahn sequential accuracy peaks at 69.4% with 200 steps per frame, then falls to 60.1% with 1,000. Gray–Scott peaks at 69.2% with 50 steps and falls to 64.1% with 1,000. Independent fits also improve with more steps; nevertheless, sequential fits lead at every tested Allen–Cahn budget from five steps onward and at every tested Gray–Scott budget.

These are validation curves. Their raw accuracies, 41.7% and 61.1%, should not be substituted for held-out comparators elsewhere.

F.4 Selective erasure does not identify slow modes as the memory carrier

The erasure test projects the measured sequential-minus-independent history component onto either the slowest or fastest quartile of non-null instantaneous tangent modes. It removes that component with strengths chosen to match the resulting change in decoded outputs. The classifier is held fixed.

Mean slow-minus-fast accuracy damage is 1.04 points. Slow-mode erasure is more damaging in three of nine cells, with five ties and one opposite ordering. The decoded-distortion ratio is 1.001, indicating that the output-matching control succeeds. The result therefore does not identify the frozen slow quartile as the useful-memory carrier, and it does not invalidate the retention rule.

F.5 Differentiating through multiple Adam fits: numerical limits

The finite-horizon analysis differentiates through the actual fresh-state Adam fitting blocks, composing the derivatives across frames. Its goal is to find endpoint directions most sensitive to observations several frames earlier, rather than using only the current tangent kernel.

One-frame replay relative parameter error is 1.9×10^{-7} on Allen–Cahn and 3.6×10^{-8} on Gray–Scott, and the Jacobian-vector/vector-Jacobian adjoint check is 1.0×10^{-14}, supporting the implementation at one frame. At eight frames, replay error rises to 4.69×10^{-5} and the estimated leading singular value shows a drift of 0.264. We therefore do not perform the intended causal erasure intervention because the composed derivative is not numerically stable enough for that test. This leaves the causal question unanswered; it is not evidence that finite-horizon memory directions do not exist.

Table 13: **Full controlled-loop diagnostics.** Local medians and slopes use nine base-field/initialization combinations at the two smallest loop scales, 0.005 and 0.01 (18 combination/scale records); figure-eight suppression uses the available control records at these scales. The minimum measured tangent-kernel change covers all nine combinations and all four scales.

Diagnostic	Ideal	Result
Median complete-field relative closure (fine RK4)	0	2.99×10^{-14}
Closure reduction, coarse→fine integration	large	$16.5\times$
Net parameter-change local slope in loop scale	2	2.023
Tangent-kernel-change local slope	2	1.979
Forward/reverse parameter cosine	-1	-0.999954
Direct Lie-bracket cosine	$+1$	0.9948
Kernel-response prediction cosine	$+1$	0.9965
Tangent-score response Spearman	$+1$	0.9903
Equal-area figure-eight suppression	large	$7.4 \times 10^{9}\times$
Minimum tangent-kernel change (all combinations/scales)	> 0	0.354
Constant-Jacobian control, kernel change	0	exactly 0

G CONTROLLED LOOPS: IMPLEMENTATION, DIAGNOSTICS, AND BOUNDARIES

G.1 EXACT TRANSPORT AND NUMERICAL ELIGIBILITY

The construction uses complete 12×12 periodic output grids and a direct QR solve, avoiding normal equations.

$$R^\top v = \dot{y}, \qquad \dot{\theta} = Qv.$$

This is the minimum-norm parameter velocity when J has full row rank. It uses no damping, singular-value truncation, or normal equations. An untruncated reduced singular-value decomposition independently checks the same parameter velocity.

The experiment uses three base fields crossed with three initializations, fixed loop-direction pairs, and scales $\{0.005, 0.01, 0.02, 0.04\}$. Fourth-order Runge–Kutta integration is run at coarse and fine resolutions in double precision on a single CPU thread. The parameter metric is the identity. "Exact" refers to the transport rule and full sampled-output constraint; numerical integration accuracy is separately measured.

All nine base-field/initialization combinations satisfy the prerequisites: rank $144/144$, maximum Jacobian condition number 1.53×10^5, base QR residual at most 6.6×10^{-16}, QR/SVD parameter-velocity disagreement at most 4.1×10^{-12}, and transport solve residual at most 1.1×10^{-15}.

G.2 FULL LOOP DIAGNOSTICS AND REPRODUCTION

The output-loop error decreases with finer integration while nonzero parameter and kernel changes remain. Across all 36 loops at the four scales, the median fine-integration relative closure is 1.71×10^{-12} and the maximum is 4.64×10^{-9}; the smaller value in Table 13 summarizes the two smallest scales. The remaining diagnostics in Table 13 show the expected area scaling, orientation reversal, figure-eight suppression, and agreement with local Lie-bracket, kernel-response, and tangent-score predictions. The constant-Jacobian decoder has exactly zero kernel change. Together, these tests support curvature of the nonlinear parameterization as the source of the controlled effect. They establish equality of all 144 sampled outputs, not equality of the continuous function away from those coordinates.

G.3 COMPARISON WITH ORDINARY FITTING

The optimizer comparison fits the same loop targets with SGD and Adam variants. Stateless SGD uses learning rate 0.009 and 100 steps per target, selected on separate validation base/initialization combinations for one-way target tracking. The evaluation uses 32 targets per loop side on six held-out base/initialization combinations at all four scales.

Practical and exact kernel-change norms correlate (0.9148), but their directions do not: the median cosine is 0.0247. Forward/reverse kernel-change cosines remain positive for SGD (+0.6676), fresh Adam (+0.9071), and carried Adam (+0.9654), so the orientation reversal characteristic of the exact transport is absent. Increasing target density from 4 to 32 reduces tracking MSE from 1.6×10^{-6} to 2.7×10^{-7}, but changes the exact-kernel cosine only from -0.222 to +0.105.

A zero-loop control repeatedly fits the unchanged starting field for 128 fresh blocks per base/initialization combination and produces exactly zero loss, weight change, and kernel change. Ordinary fitting therefore fails the characteristic exact-transport signatures in this tested setting. This is the relevant boundary on using the construction to explain the practical sequential encoder.

The controlled-loop experiment therefore establishes output-preserving changes in tangent geometry, but it does not show that ordinary SGD/Adam reproduces the transport or that such geometry edits improve subsequent learning.