IN-CONTEXT LEARNING ACROSS THE LOGISTIC FAMILY

Anonymous authors
Paper under double-blind review

ABSTRACT

Prior theoretical studies of in-context learning (ICL) in transformers have formulated task inference in a regression setting, where tasks are weight vectors in parameter space and the data are independently sampled from a known distribution. In contrast, time series prediction involves observations generated from an evolving latent system that can exhibit temporal dependencies while the available representation may only preserve part of the underlying state. We study ICL in transformers trained to perform next token prediction on trajectories generated by the logistic family. Using the logistic family as a controlled setting, we investigate how training-task diversity, context length, and observation resolution affect model performance while retaining the latent structure of the task. At a fixed trajectory budget, generalization to new maps emerges sharply beyond a critical level of training-task diversity. The resulting generalization structure depends on representation: finer resolutions require greater task diversity to generalize but attain lower limiting error. Under restricted training support, transfer is strongly asymmetric. Models trained on intervals containing chaotic dynamics often predict lower-complexity systems accurately, whereas reverse transfer is poor. Our results show that in-context generalization depends jointly on task diversity, observation representation, and the dynamical regimes represented during training.

1 INTRODUCTION

Transformers pretrained on large collections of time series data can forecast previously unseen time series, given only a short trajectory as context (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024). This ability extends to chaotic dynamical systems (Zhang & Gilpin, 2025; Lai et al., 2026) and is a form of in-context learning (ICL), in which a model adapts to a new task from the examples in its context without updating its parameters (Brown et al., 2020). How these models generalize is only beginning to be understood: transformers trained on a single dynamical system can forecast others by estimating transfer operators in context (Bao et al., 2026), yet a baseline that simply copies matching segments of the context outperforms leading time-series foundation models on many dynamical systems (Zhang & Gilpin, 2026). A basic question is what the pretraining data must contain in order for in-context learning to emerge. How many distinct systems must a model see during training, and which ones?

A significant body of work on in-context learning in transformers formulated task inference in a regression setting, where tasks belong to a predefined parameter space and the data are independently sampled input-output pairs from a known distribution. Transformers trained on a small number of distinct tasks specialize to them, whereas beyond a threshold level of task diversity they learn a solution that generalizes to unseen tasks (Raventós et al., 2023; Lu et al., 2025), including tasks outside the region of task space seen during training (Goddard et al., 2025).

Forecasting a dynamical system differs from regression in three ways. First, rather than a set of independent examples, a single trajectory is presented in context and observations within this trajectory can be temporally dependent. Thus, the underlying dynamics determines what evidence about the system is available in context. For example, a trajectory that settles onto a fixed point visits only a few states, whereas a trajectory from a chaotic system can explore a much larger region of phase space.

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.

Second, the system is often observed through a finite representation, such as the uniform bins some foundation models use to tokenize real values (Ansari et al., 2024). Third, dynamical systems can realize a wide range of behaviors, from fixed points and periodic orbits, to quasi-periodic motion and chaos. Furthermore, systems that are nearby in parameter space can produce qualitatively different behaviors (i.e., on either side of a bifurcation). How these features affect the task diversity necessary for generalization and which unseen systems a model can predict as a function of its training coverage remain open questions.

Why the logistic family? We study these questions using the logistic family,

$$f_r(x) = rx(1-x), \qquad x \in [0,1], \quad r \in [0,4],$$

a canonical model of one-dimensional nonlinear dynamics (May, 1976; de Melo & van Strien, 1993) that contains a wide range of dynamical behavior while being governed by a single parameter. For $r < 1$, trajectories converge to the fixed point at zero, thus we exclude $r < 0.5$, and retain $[0.5, 1)$ to represent this regime. For $1 < r < 3$, trajectories converge to the nonzero fixed point $x^*(r) = 1 - 1/r$ and increasing r beyond 3 eventually produces a sequence of period-doubling bifurcations leading to the emergence of chaotic dynamics (Figure 1**(b)**). Thus, changing r changes the attractor and the regions of state space visited by typical trajectories.

For our purposes, each $r \in [0, 4]$ specifies a unique task f_r that generates trajectories for the model to predict. Models can thus be trained on trajectory data from a subset of the logistic family and evaluated on a new trajectory generated by either a previously observed or unseen task. To predict the next observation, the model must use the trajectory data provided in context to infer the latent dynamics. Because the generating system determines which states are visited, the evidence available for identifying the underlying task depends on the how much data is given in context. Another determining factor is the observation resolution. Instead of using continuous values, we partition $[0, 1]$ into N bins producing a token sequence $\{t_n\}$ that is then fed to the model. Thus, a deterministic map induces a resolution-dependent symbolic process. Coarse partitions merge dynamically distinct states, while finer partitions preserve more state information but produce a larger and more sparsely sampled transition space. This allows for careful study of the relationship between task diversity, context length, and observation resolution and how they combine to impact transformer prediction and generalization. This interaction is generally difficult to isolate in natural time series data, where the latent mechanism, observation process, and training distribution are generally unknown or difficult to characterize fully.

We find that generalization to unseen maps emerges as the number of distinct training maps exceeds a threshold, while error on seen maps remains approximately unchanged despite the decreasing number of trajectories available per map (Figure 2). The location of this crossover depends strongly on the observation representation. Finer tokenizations require more training maps to generalize but attain lower error at high diversity, whereas coarser tokenizations generalize with fewer maps but plateau at higher error. Increasing the vocabulary through interchangeable synonym tokens does not produce a comparable shift, indicating that the effect is driven by the additional state information represented by finer bins rather than by vocabulary size (Figure 3). Context length, in contrast, has little effect on the crossover. All context lengths studied reach the same high-diversity error floor, and only the shortest contexts cross over noticeably earlier. Finally, when training is restricted to parameter subintervals, models achieve low error within their training support but exhibit structured and asymmetric transfer outside it. Models trained on intervals that contain chaotic dynamics often predict simpler systems accurately, whereas the reverse transfer is poor, and models trained just above the first period-doubling bifurcation accurately predict a wide range of fixed-point systems whose fixed points coincide with states on their training orbits (Figure 4). Together, these results show that generalization across dynamical systems depends jointly on training-task diversity, observation resolution, and the dynamical regimes represented during training.

2 Related Work

Task diversity and in-context generalization. Our task-diversity experiments build on Raventós et al. (2023), who showed that for linear regression tasks, transformers undergo a phase transition in their generalization behavior as the number of tasks seen during pretraining increases. Goddard et al. (2025) studied the out-of-distribution version of this question, finding that increasing task

dissimilarity during pretraining drives a similar transition in the OOD generalization behavior of the model. Similar transitions driven by the number or diversity of training tasks occur in modular arithmetic (He et al., 2024), in-context classification (Chan et al., 2022; Reddy, 2024; Nguyen & Reddy, 2024), and meta-learned in-context learners (Kirsch et al., 2022), and have been analyzed theoretically for linear attention (Wu et al., 2024; Lu et al., 2025; Letey et al., 2026b). Other work has characterized the algorithms that trained transformers implement in context (Garg et al., 2022; Akyürek et al., 2023; von Oswald et al., 2023; Ahn et al., 2023; Bai et al., 2023) and the limits of their generalization beyond the function classes seen during pretraining (Yadlowsky et al., 2023). Nearly all of these settings treat a context as a set of independent input–output pairs; Letey et al. (2026a) recently showed that sequential correlations reduce the effective context length of in-context regression.

In-context learning of sequential and dynamical processes. Transformers trained on sequences from random Markov chains learn to estimate transition statistics in context, passing through a unigram stage before forming statistical induction heads (Edelman et al., 2024). For continuous-valued sequences, transformers learn linear dynamical systems and autoregressive processes in context (Li et al., 2023; Sander et al., 2024; Cole et al., 2025), and can infer solution operators of differential equations from example pairs (Yang et al., 2023). Pretrained language models also infer the transition rules of dynamical systems, including the logistic map, from their context, with accuracy that improves as the context grows (Liu et al., 2024). Closest to our work, Bao et al. (2026) train small transformers on trajectories of a single dynamical system and find that they can forecast trajectories from other dynamical systems. Our work differs by training on a parametric family, which lets us vary task diversity and observation resolution.

Forecasting dynamical systems with pretrained models. Time-series foundation models pretrained on large, heterogeneous corpora forecast previously unseen series zero-shot (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024; Rasul et al., 2023; Goswami et al., 2024), and language models pretrained on text can forecast numerical series written as digit strings (Gruver et al., 2023). These models differ in how they represent real values: Chronos scales each series and quantizes it into uniform bins (Ansari et al., 2024), much as we do, whereas other models embed patches of continuous values (Das et al., 2024; Woo et al., 2024). Such models can forecast chaotic systems zero-shot (Zhang & Gilpin, 2025), and models pretrained on large synthetic collections of chaotic systems generalize to unseen ones (Lai et al., 2026; Hemmer & Durstewitz, 2025).

3 Contributions

Our contributions are as follows:

1. We introduce a controlled setting for in-context learning of dynamical systems based on the logistic family $\{f_r\}$, in which the number of training maps, context length, observation resolution, and training support can be varied independently, and model behavior can be explained by known properties of each map.
2. We find a clear threshold in the number of training maps beyond which models generalize to unseen maps, as in the regression setting. Finer observation resolution moves this threshold to more maps but lowers the error attained beyond it. Increasing vocabulary size at fixed resolution does not move it, and context length leaves the attainable error unchanged.
3. We train models on restricted intervals of the family and show that out-of-support transfer is asymmetric and can be nonlocal: intervals containing chaotic dynamics transfer broadly, and a simple state-overlap calculation predicts where models trained just above the first period-doubling bifurcation transfer to fixed-point systems.

4 Methods

ICL of Dynamical systems: Consider the logistic family

$$f_r(x) = rx(1-x), \qquad x \in [0,1], \quad r \in [0.5, 4]. \tag{1}$$

A task is specified by a parameter value r, and a trajectory from that task evolves according to

$$x_{n+1} = f_r(x_n). \tag{2}$$

All observations within a context are generated by the same latent map f_r. In our experiments, the map parameter r is either sampled iid from Uniform$([a, b])$ with $a, b \in [0.5, 4]$, or restricted to a finite set $\mathcal{R}_{\text{train}} = \{r_1, \ldots, r_m\}$ where we call m the number of training *tasks*. For a typical initial condition, the long-run visitation frequency defines the physical invariant measure

$$\mu_r(A) = \lim_{T\to\infty} \frac{1}{T} \sum_{n=0}^{T-1} \mathbf{1}\{x_n \in A\},$$

when this limit exists. For example, at $r = 4$ the logistic map has the invariant density

$$\rho_4(x) = \frac{1}{\pi\sqrt{x(1-x)}},$$

so typical long trajectories visit states near 0 and 1 more frequently than states near the center of the interval, as shown in Figure 1**(c)**. Sensitivity to perturbations is characterized by the Lyapunov exponent which is given by

$$\lambda(r) = \lim_{T\to\infty} \frac{1}{T} \sum_{n=0}^{T-1} \log |f_r'(x_n)| = \lim_{T\to\infty} \frac{1}{T} \sum_{n=0}^{T-1} \log |r(1-2x_n)|$$

as shown in Figure 1**(d)**. When $\lambda(r) < 0$, typical trajectories converge to an attracting cycle, after which the token sequence is periodic and the next token is determined by the preceding ones. When $\lambda(r) > 0$, the orbit does not settle onto a cycle, and a small uncertainty in the state grows on average by a factor of $e^{\lambda(r)}$ per step.

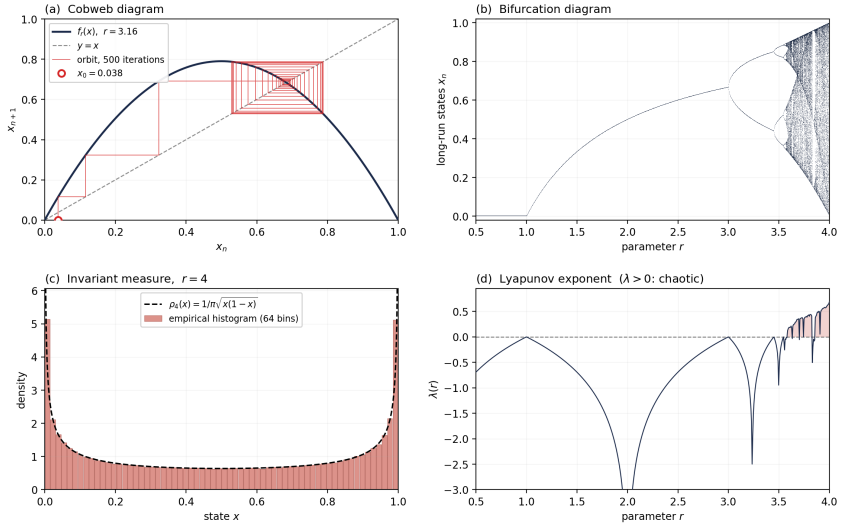

Figure 1: **Properties of the logistic family. (a)** Cobweb diagram at $r = 3.16$ from $x_0 = 0.038$ (500 iterations, red): the orbit climbs from near the origin, is repelled by the unstable fixed point $x^* = 1 - 1/r \approx 0.684$ where f_r meets the diagonal, and converges onto the period-2 cycle. **(b)** Bifurcation diagram showing the periodic windows, period-doubling cascade and the chaotic band. **(c)** Invariant measure at $r = 4$: an empirical histogram of 3×10^5 visited states at the 64-bin resolution, against the analytic arcsine density $\rho_4(x) = 1/\big(\pi\sqrt{x(1-x)}\big)$. **(d)** Lyapunov exponent $\lambda(r)$ on the same r axis as (b), with $\lambda = 0$ marked and $\lambda > 0$ shaded; periodic windows in (b) coincide with λ dipping below zero, and $\lambda(4) = \ln 2$.

Trajectory generation and tokenization. For each trajectory, we choose a value of r, sample $x_0 \sim \mathrm{Unif}(0.05, 0.95)$ and iterate the map f_r starting at x_0. The same initial-condition distribution is used across tasks and experimental conditions. For all experiments, each trajectory begins at x_0 and therefore contains both transient and long-run behavior. We include transient observations since they may provide additional information about the generating map f_r. Our experiments thus do not sample trajectories directly from the invariant measure. We then convert this real-valued trajectory $(x_0, x_1, \dots)$ in $[0, 1]$ to a symbolic sequence by partitioning $[0, 1]$ into N equal bins. To do so, we define a uniform N-bin tokenizer by

$$Q_N(x) = \min\{\lfloor Nx \rfloor, N-1\}, \qquad Q_N : [0,1] \to \{0, \dots, N-1\}. \tag{3}$$

The clipping in (3) assigns the endpoint $x = 1$ to the final bin. The resulting token sequence is $\{t_n\} = Q_N(\{x_n\})$. We train models on tokenized sequences $(t_0, t_1, \dots) \in \{0, \dots, N-1\}^{\mathbb{N}}$. We choose uniform bins to study the effects of input resolution on prediction and generalization.

Note that uniform bins generally do not preserve the first-order structure of the underlying map. For each map, the state x_{n+1} depends only x_n, but two states assigned to the same token can fall into different output bins after applying f_r. A given token t_n corresponds to a single bin and thus the preceding token sequence can further constrain the location of x_n within t_n. It follows that the distribution of t_{n+1} may depend on more tokens than t_n, even though the underlying map is deterministic and first order. Thus, tokenization affects both the difficulty of one-step prediction and the value of additional context. Coarse tokenizations collapse many continuous states onto the same token, while finer tokenizations preserve more information but create a larger set of tokens and transitions between tokens.

Note that we apply this variable tokenization to the model's inputs only. We study two different choices for the model's objective function. In the first, we ask the model to predict one of 64 output tokens, each corresponding to a subinterval of $[0, 1]$. In this case, we train models with cross-entropy loss (and thereby penalize all prediction errors equally, no matter how wrong the model is). In the second, we ask the model to predict a real number on the output, and train models using the square loss (penalizing predictions far from the ground truth more heavily).

Model architecture and training. We train causal (decoder-style) transformers to do next-token prediction, $\hat{p}(t_{n+1} \mid t_{n-L+1}, \dots, t_n) \in \mathbb{R}^{64}$. Tokens are mapped to $\mathbb{R}^d$ by learned embeddings.

We use learned positional embeddings, pre-norm transformer encoder layers with a causal mask, and a linear readout to either one of 64 bins (for cross-entropy loss) or a scalar (for square loss). If not otherwise specified, we use $N = 64$ bins, context length $L = 50$, $d_{\mathrm{model}} = 128$, 4 attention heads, 4 layers, feedforward width $d_{\mathrm{ff}} = 4d_{\mathrm{model}}$, and dropout 0.1, totaling $\approx 8.2 \times 10^5$ trainable parameters. We train with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}), initial learning rate 3×10^{-4} under a cosine schedule, gradient clipping at 1.0, and batch size 256.

Evaluation. We measure new-task error on a 300-point grid over $r \in [0.5, 4]$ and seen-task error on fresh trajectories at training parameters. At each parameter, we average the loss (cross-entropy or square loss) over 30 trajectories with independent initial conditions, then average across parameters.

Terminal-position prediction objective. Each training example consists of a context of L observations followed by a single one-step prediction target. Although the transformer produces a hidden representation at every context position, the training loss is applied only to the prediction made from the final position:

$$\mathcal{L}_{\mathrm{CE}} = -\log \hat{p}_\theta\big(t_{n+1} \mid t_{n-L+1:n}\big)$$

for categorical models, and

$$\mathcal{L}_{\mathrm{MSE}} = \big(\hat{x}_{\theta,n+1}(t_{n-L+1:n}) - x_{n+1}\big)^2$$

for scalar-output models. We do not apply auxiliary next-token losses at the earlier positions in the context, as is conventional in dense autoregressive language-model training.

5 Experimental Results

Since each $r \in [0.5, 4]$ specifies a unique dynamical system f_r in the logistic family, a natural question to consider is how many examples of such systems does a model need to observe during

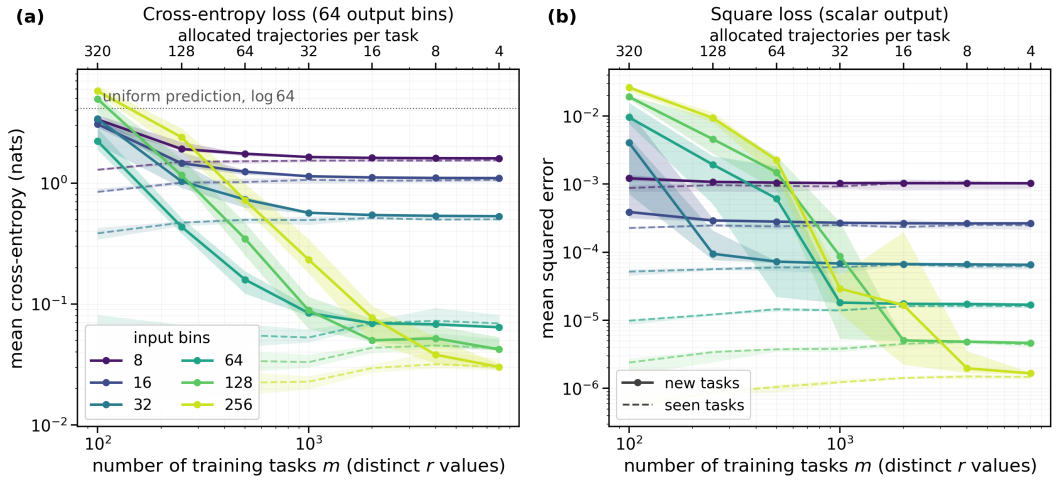

Figure 2: **Input resolution shifts the task-diversity threshold.** Models are trained on $32{,}000$ trajectories split across m maps and evaluated at held-out (solid) and training (dashed) parameters. As in in-context regression, new-task error drops sharply above a critical m while seen-task error stays flat. Finer input partitions move the transition to larger m but reach lower error at high diversity. **(a)** cross-entropy, 64 output bins (dotted: $\log 64$). **(b)** MSE, scalar-output models. Medians over four seeds; seeds can straddle the square-loss crossover. Models with 8–32 input bins are evaluated at 160k steps, 64–256 at 320k.

pre-training in order to generalize to unseen systems? Furthermore, how does this number depend on the resolution of the pretraining data? Following Raventós et al. (2023), we vary task diversity by training on a finite pool of $m \in \{100, 250, 500, 1000, 2000, 4000, 8000\}$ maps. We fix the dataset at $32{,}000$ trajectories, so increasing m *reduces* the number of trajectories per map from 320 to 4. Moving from linear regression to dynamical systems allows us to investigate how the model's generalization ability changes as the resolution of the input data varies over $N_{\text{in}} \in \{8, 16, 32, 64, 128, 256\}$ bins.

Figure 2 demonstrates analogous phenomena to that of Raventós et al. (2023) in our dynamical setting. Below a critical number of training maps, models predict seen parameters well but fail on new ones; above it, new-task error drops sharply toward seen-task error. Seen-task error is nearly independent of m, even though the number of trajectories per map falls by a factor of 80. A transition is not guaranteed here. Existing accounts assume that the inputs in context are drawn independently of the task (Wu et al., 2024; Lu et al., 2025). The location of this transition depends strongly on input resolution.

At $m = 100$ in Figure 2**(a)**, intermediate resolutions generalize best, with the 64-bin input model performing best and the 256-bin input model performing worst. Similarly for the regression models in Figure 2**(b)**, the 16-bin input model performs best and the 256-bin input model performs worst. As m increases, this ordering reverses for both model types. At $m = 8000$, finer partitions produce progressively lower error, with the 256-bin model performing best.

These results identify a tradeoff between observation resolution and the task diversity required for generalization. Coarse tokenizations impose a higher limiting error, but their new-task performance stabilizes with fewer training systems relative to models with finer tokenizations. Fine tokenizations preserve more state information and achieve substantially lower error at high diversity.

In Figure 3**(a)**, we vary the context length L used for training and evaluation. Because the loss is applied only at the final position, L is the only context length at which each model is trained. At high diversity, all context lengths reach approximately the same new-task error floor, so additional context does not improve the best attainable performance. The crossover is also largely insensitive to L, with one exception: at small m, the $L = 10$ models have lower new-task error than longer-context models and reach the common low-error regime by $m \approx 250$, whereas most longer-context models reach it only at larger m.

For $L > 10$, the crossover varies across seeds with no consistent trend in L. Surprisingly, longer context does not seem to help performance.

In Figure 3**(b)**, we test whether the resolution-dependent transition in Figure 2 is a direct consequence of increasing the spatial resolution, or a side effect of increasing the input vocabulary. To investigate this, we increase the vocabulary size while holding resolution fixed, introducing "synonym" tokens for each bin. Increasing the spatial resolution from 16 to 256 bins moves the estimated transition point m^* from approximately 194 to 831 training maps. In contrast, splitting the original 64 bins into interchangeable synonym tokens increases the vocabulary from 64 to 1024 while leaving m^* approximately constant. This indicates that the movement of the transition point is associated with the state distinctions encoded by the tokenization rather than vocabulary size alone.

Context length and observation resolution therefore affect generalization in different ways. Longer context leaves the high-diversity error floor unchanged and, apart from the shortest contexts, has little effect on the crossover. Finer observation resolution also increases the diversity required for generalization, but ultimately enables substantially lower prediction error. In our experiments, coarse tokenization imposes an error floor that neither additional task diversity nor longer context can overcome.

Increasing m changes how densely the family is sampled but not which dynamical regimes are represented. To examine the role of training support, we restrict training to parameter subintervals and evaluate each model across the full family (Section 5.1).

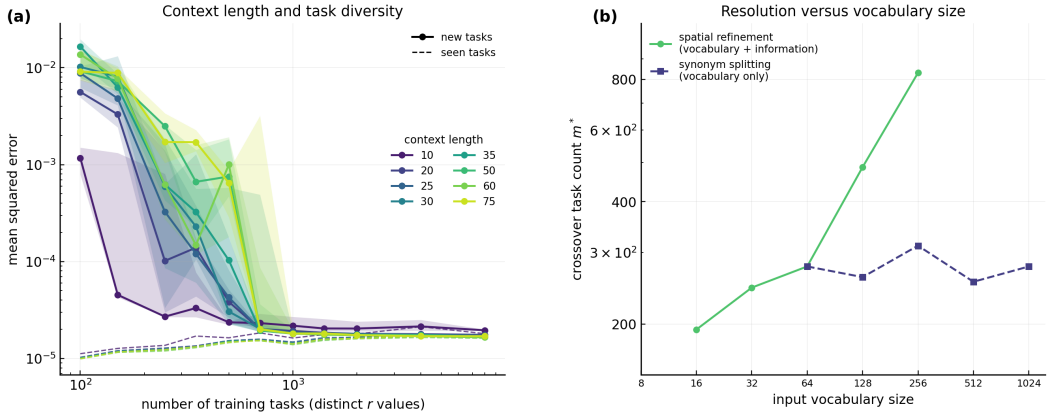

Figure 3: **Context length leaves the error floor unchanged. (a)** Mean squared error versus the number of training maps for models trained at different context lengths. Solid curves denote new tasks and dashed curves denote fresh trajectories at training parameter values. Lines show medians over three seeds and shading shows the full seed range. All models use 64 input bins, 32,000 trajectories, and 320k training steps; setting trajectory length to $L + 100$ provides 100 windows per trajectory. New-task error decreases sharply with diversity at every context length, with short-context models often reaching the common low-error regime at smaller m. **(b)** Crossover task count m^*, defined by the logarithmic midpoint between new-task error at $m = 100$ and its large-m plateau. Increasing spatial resolution moves the crossover to larger m, whereas increasing vocabulary size through interchangeable synonym tokens produces no comparable shift. Panel **(b)** shows one seed per condition.

5.1 Restricted Training Support Produces Structured Out-of-Support Transfer

Restricted-window protocol. Our previous analysis demonstrated that increasing the diversity of tasks shown during training allows the model to generalize to new systems not seen at training time. However, all tasks are not created equal and thus specific parameter regions of $r \in [0.5, 4]$ may have an outsized effect on generalization to new systems.

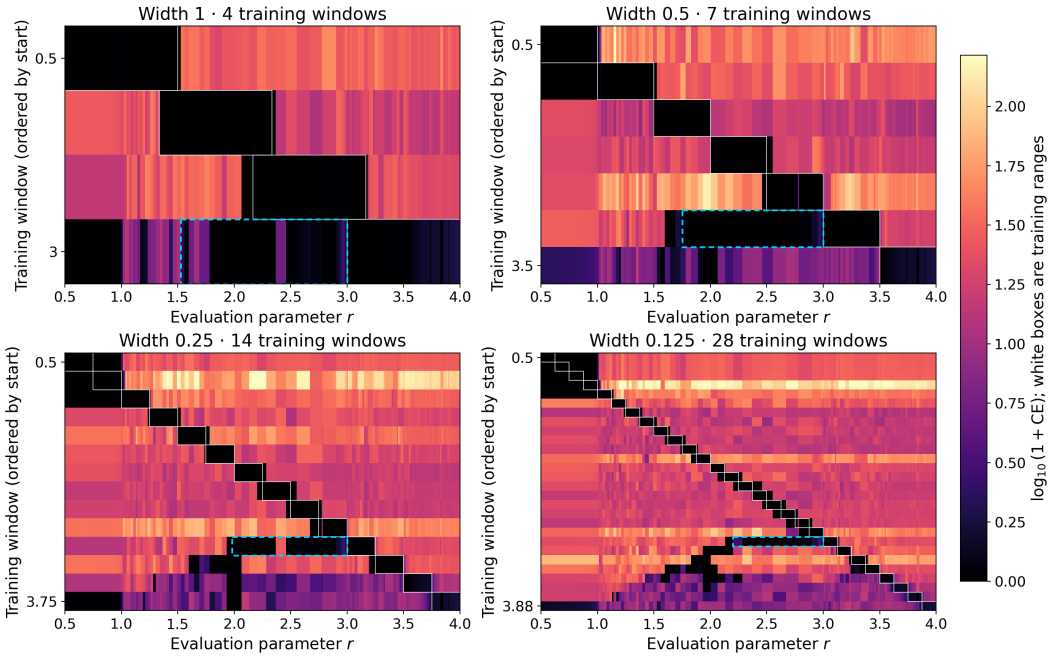

Figure 4: **Out-of-support transfer is asymmetric.** Rows: transformers trained on $W = [a, a + w]$ (white rectangles). Columns: evaluation parameters on a 300-point grid over $r \in [0.5, 4]$. Panels: $w \in \{1, 0.5, 0.25, 0.125\}$. Color: $\log_{10}(1 + \mathrm{CE})$. Error generally rises outside the training interval, but not uniformly with distance. Models trained on intervals containing chaotic dynamics often predict regular systems accurately, whereas models trained on regular intervals transfer poorly to chaotic ones. Models trained on $[3, 3 + w]$ predict fixed-point systems over $[r_{\min}(w), 3]$, the range predicted by state overlap (cyan). All models: 8,000 trajectories, 64 input and output bins, $L = 50$, final 160k-step checkpoint.

To investigate this, we train transformers on subsets of the full parameter space:

$$W = [a, a + w] \subset [0.5, 4].$$

We consider widths $w \in \{1, 0.5, 0.25, 0.125\}$ and distribute respectively $4, 7, 14$, and 28 intervals across the full parameter range, producing 53 models. The intervals of width 1 overlap slightly, while the remaining intervals tile the range. Each model sees 8,000 trajectories during training, and we hold the density of distinct training parameters per unit r approximately constant. Intervals of width $1, 0.5, 0.25$, and 0.125 therefore contain $2286, 1143, 571$, and 286 randomly sampled training parameters, respectively. All models use 64 input and output bins, context length $L = 50$, and are trained for $160k$, training steps. We evaluate the final checkpoint on fresh trajectories from a common 300-point grid spanning $r \in [0.5, 4]$.

Figure 4 shows that all models predict well within the parameter interval seen during training and that the error generally increases sharply once the evaluation parameter leaves the interval. However, several models exhibit near zero loss regions that are separated from their training intervals, and similar structures recur at all four window widths.

A recurring state-overlap pattern. A consistent prediction pattern occurs for the row trained on $W = [3, 3 + w]$ in each panel. These models predict a large interval of fixed-point systems with $r < 3$ substantially better than nearby models whose training windows do not begin at $r = 3$. The left boundary of this region can be predicted from a simple calculation. The attracting fixed point for $1 < r < 3$ is $x^*(r) = 1 - \frac{1}{r}$. For $R > 3$, the lower branch of the period-two orbit (Figure 1**(a)**) is

$$x_-(R) = \frac{R + 1 - \sqrt{(R - 3)(R + 1)}}{2R}.$$

Setting $R = 3 + w$ and matching $x^*(r_{\min}) = x_-(R)$ gives

$$r_{\min}(w) = \frac{1}{1 - x_-(3 + w)}.$$

The cyan rectangles in Figure 4 mark $r \in [r_{\min}(w), 3]$ for the model trained on $W = [3, 3 + w]$. Over this interval, the target fixed points sweep the same range of state values as the lower period-two branch across the training window. The agreement with the observed low-loss region suggests that overlap in represented states helps explain this otherwise nonlocal transfer.

The corresponding scalar-output experiment is shown in Figure 7 of the Supplementary Material. It exhibits the same broad contrast between low in-support error and substantially larger out-of-support error, together with structured regions of nonlocal transfer. Thus, the phenomenon is not specific to cross-entropy or to discrete output space.

6 Discussion and future work

We study in-context learning in transformers trained to predict tokenized trajectories of the logistic family, varying task diversity, observation resolution, context length, and training support independently. Our results suggest that generalization across dynamical systems depends on how many systems are seen during training, on which states and transitions their trajectories visit, and on how finely the tokenization distinguishes those states.

Finer tokenizations preserve more state information and reach lower error at high diversity, but they also resolve more distinct states and transitions, and more diverse training data is needed before the model generalizes. Context length, by contrast, has little effect at high diversity. All context lengths achieve the same minimum error, even though longer contexts carry more evidence about the generating map.

Which dynamical regimes are represented in training has a substantial impact on model performance. Models trained on intervals containing chaotic dynamics often predict simpler systems accurately, whereas models trained on non-chaotic intervals predict chaotic systems poorly. These observations are consistent with a mix of strategies: once a trajectory settles onto a cycle, copying earlier tokens suffices, as with in-context parroting (Zhang & Gilpin, 2026), whereas chaotic trajectories require applying the local action of the map, as in accounts of in-context transfer-operator estimation (Bao et al., 2026). For time-series foundation models, this suggests that the tokenization and the pretraining data selection choices should informed one another, and that systems whose trajectories explore more of their state space may yield broader transfer than additional systems from simpler regimes.

Limitations and future work. We studied a single one-parameter family of deterministic one-dimensional maps, with uniform partitions, noise-free observations, and small transformers trained with a loss on the final context position only. Natural extensions include other unimodal maps, higher-dimensional systems in which the context must reconstruct unobserved state, noisy observations, and non-uniform or learned tokenizations. Finally, a theory of how the critical number of training maps scales with resolution could connect these results to task-diversity thresholds derived for linear attention (Lu et al., 2025).

AI use statement

In this work, we used generative AI tools to implement experiments and to assist with editing text. All AI-assisted code was reviewed and tested by the authors. We take responsibility for the final content of this work.

Ethics statement

This work involves no personal or sensitive data, or third-party datasets. The models were trained with modest compute. We do not foresee direct harmful applications.

Reproducibility statement

The data in this work are generated procedurally from the dynamical systems specified in Section 4. The data-generation, model, training, and evaluation protocols are described in the main text and supplemental material, including the principal hyperparameters and training budgets. Code will be released when the author list is made public.

References

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations*, 2023.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems*, volume 36, pp. 57125–57211, 2023.

A. Bao, J. Lai, and W. Gilpin. Transformers for dynamical systems learn transfer operators in-context. *arXiv preprint arXiv:2602.18679*, 2026.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Frank Cole, Yulong Lu, Tianhao Zhang, and Yuxuan Zhao. In-context learning of linear dynamical systems with transformers: Error bounds and depth-separation. *arXiv preprint arXiv:2502.08136*, 2025.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 2024.

Welington de Melo and Sebastian van Strien. *One-Dimensional Dynamics*. Springer, Berlin, 1993.

Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning Markov chains. In *Advances in Neural Information Processing Systems*, volume 37, pp. 64273–64311, 2024.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598, 2022.

Chase Goddard, Lindsay M. Smith, Vudtiwat Ngampruetikorn, and David J. Schwab. When can in-context learning generalize out of task distribution? In *International Conference on Machine Learning*, 2025.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *International Conference on Machine Learning*, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Tianyu He, Darshil Doshi, Aritra Das, and Andrey Gromov. Learning to grok: Emergence of in-context learning and skill composition in modular arithmetic tasks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 13244–13273, 2024.

Christoph Jürgen Hemmer and Daniel Durstewitz. True zero-shot inference of dynamical systems preserving long-term statistics. *arXiv preprint arXiv:2505.13192*, 2025.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. General-purpose in-context learning by meta-learning transformers. *arXiv preprint arXiv:2212.04458*, 2022.

Jeffrey Lai, Anthony Bao, and William Gilpin. Panda: A pretrained forecast model for chaotic dynamics. In *International Conference on Learning Representations*, 2026.

Mary Letey, Yue M. Lu, Cengiz Pehlevan, and Jacob A. Zavatone-Veth. Sequential correlations change in-context learning: Effective context length and architectural mismatch. *arXiv preprint arXiv:2607.03660*, 2026a.

Mary Letey, Jacob A. Zavatone-Veth, Yue M. Lu, and Cengiz Pehlevan. Pretrain-test task alignment governs generalization in in-context learning. In *International Conference on Learning Representations*, 2026b.

Yingcong Li, M. Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, 2023.

Toni J. B. Liu, Nicolas Boullé, Raphaël Sarfati, and Christopher J. Earls. LLMs learn governing principles of dynamical systems, revealing an in-context neural scaling law. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15097–15117, 2024.

Yue M. Lu, Mary Letey, Jacob A. Zavatone-Veth, Anindita Maiti, and Cengiz Pehlevan. Asymptotic theory of in-context learning by linear attention. *Proceedings of the National Academy of Sciences*, 122(28):e2502599122, 2025.

Robert M. May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560): 459–467, 1976.

Alex Nguyen and Gautam Reddy. Differential learning kinetics govern the transition from memorization to generalization during in-context learning. *arXiv preprint arXiv:2412.00104*, 2024.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, et al. Lag-Llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *International Conference on Learning Representations*, 2024.

Michael E. Sander, Raja Giryes, Taiji Suzuki, Mathieu Blondel, and Gabriel Peyré. How do transformers perform in-context autoregressive learning? In *International Conference on Machine Learning*, 2024.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174, 2023.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, 2024.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L. Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? In *International Conference on Learning Representations*, 2024.

Steve Yadlowsky, Lyric Doshi, and Nilesh Tripuraneni. Pretraining data mixtures enable narrow model selection capabilities in transformer models. *arXiv preprint arXiv:2311.00871*, 2023.

Liu Yang, Siting Liu, Tingwei Meng, and Stanley J. Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023.

Yuanzhao Zhang and William Gilpin. Zero-shot forecasting of chaotic systems. In *International Conference on Learning Representations*, 2025.

Yuanzhao Zhang and William Gilpin. Context parroting: A simple but tough-to-beat baseline for foundation models in scientific machine learning. In *International Conference on Learning Representations*, 2026.

A Supplementary Material

Chaotic coverage is associated with broader transfer. Across all four window widths, models trained on intervals that extend into the chaotic regime of the logistic family appear to produce lower error over a larger portion of the family. To examine this generalization phenomenon while accounting for the intrinsic difficulty of the target parameter, we compare each out-of-support prediction against models of the same window width for which that target lies inside the training interval. Let $W = [a, a + w]$ denote a training interval and let $\mathrm{CE}_W(r)$ be the mean cross-entropy obtained by the model trained on W when evaluated at parameter r. We define the transformed loss

$$\ell(W, r) = \log_{10}(1 + \mathrm{CE}_W(r)) \,. \tag{4}$$

For each target r and window width w, define the matched in-support reference

$$\ell_{\mathrm{in}}(r; w) = \underset{\substack{W' :\, |W'|=w \\ r \in W'}}{\mathrm{median}} \ell(W', r). \tag{5}$$

For an out-of-support pair $r \notin W$, we then define the excess loss

$$\Delta(W, r) = \underbrace{\ell(W, r)}_{\substack{\text{out-of-support loss} \\ \text{at target } r}} - \underbrace{\ell_{\mathrm{in}}(r; |W|)}_{\substack{\text{typical same-width in-support} \\ \text{loss at the same target } r}} . \tag{6}$$

The reference is therefore constructed from models with the same training-window width that were trained in support at the target r. This normalization removes much of the variation due to target difficulty and systematic differences between window widths. For each training window, we summarize transfer by averaging $\Delta(W, r)$ over evaluation parameters outside W.

We quantify the dynamical complexity represented within a training interval by its chaotic coverage,

$$C_{\mathrm{chaos}}(W) = \frac{1}{|W|} \int_W \mathbf{1}_{\{\lambda(r)>0\}} \, dr \tag{7}$$

evaluated numerically on a dense parameter grid. We also compute the mean out-of-support penalty separately over regular targets with $\lambda(r) \leq 0$ and chaotic targets with $\lambda(r) > 0$.

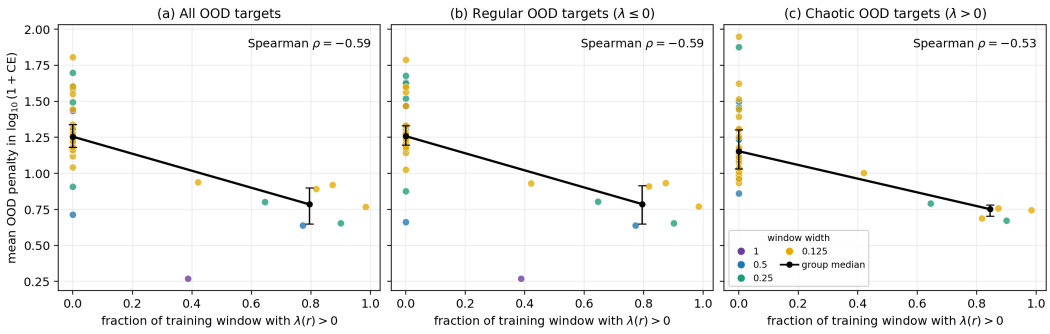

Figure 5: **Out-of-interval transfer as a function of chaotic coverage in the training interval.** Each point represents one of the 53 training windows, colored by its width. The horizontal axis is the fraction of the interval for which $\lambda(r) > 0$. The vertical axis is the mean normalized out-of-support penalty from (6). **(a)** Penalty averaged over all out-of-support targets. **(b)** Penalty averaged over regular targets with $\lambda(r) \leq 0$. **(c)** Penalty averaged over chaotic targets with $\lambda(r) > 0$. Black points and lines compare the median penalty for windows with zero and positive chaotic coverage; error bars show the interquartile range.

Greater chaotic coverage is associated with a smaller out-of-support penalty. The association is similar when transfer is averaged over all targets ($\rho = -0.59$), regular targets ($\rho = -0.59$), or chaotic targets ($\rho = -0.53$). Thus training on chaotic intervals improves prediction across all test systems.

These correlations are descriptive rather than inferential. The 53 windows overlap, only eight contain parameters with positive Lyapunov exponent, and one training seed is available for each interval. Nevertheless, the consistency across window widths and target regimes gives quantitative support to the qualitative pattern visible in Figure 4.

Represented state and transition similarity predict transfer. We next test whether the advantage of dynamically complex training intervals can be explained by the token-level processes they expose to the model. For each evaluation parameter r, let

$$J_r(i,j) = \Pr(t_n = i, t_{n+1} = j)$$

denote the represented one-step transition process under the same finite-length trajectory and tokenization protocol used for evaluation. Let

$$S_r = \{(i,j) : J_r(i,j) > 0\}$$

denote its empirical support. We define the directed transition-support distance from a target r to a possible source r' as

$$d_{\mathrm{supp}}(r, r') = \sum_{i,j} J_r(i,j) \min_{(k,l)\in S_{r'}} \frac{|i-k| + |j-l|}{N}. \tag{8}$$

This measures how far the target transition mass lies from transitions represented by the source system on the two-dimensional token-transition grid. The nearest support mismatch for a training interval is then

$$D_{\mathrm{supp}}(W, r) = \min_{r'\in W} d_{\mathrm{supp}}(r, r').$$

Transition processes are estimated using a common set of 512 initial conditions for each evaluation parameter.

We separately measure whether the target visits the same represented states as the training data. Let p_r be the token-occupancy distribution induced by the target trajectories and let p_W be the corresponding distribution pooled over the actual trajectories used to train the model. We define the context-state mismatch as

$$D_{\mathrm{state}}(W, r) = D_{\mathrm{JS}}(p_r \,\|\, p_W),$$

where context positions are weighted according to their frequency among the overlapping training examples. Finally, as a parameter-space baseline, we use

$$d_r(W, r) = \max\{a - r,\ r - (a + w),\ 0\}, \qquad W = [a, a + w].$$

At the window–target level, both representation-based quantities are associated with prediction error. Context-state mismatch has Spearman correlation $\rho = 0.56$ with the normalized out-of-support penalty, while nearest transition-support mismatch has correlation $\rho = 0.44$. In contrast, distance from the training interval in parameter space has only $\rho = 0.07$. Thus, two systems that are close in r need not induce similar represented processes, while systems separated in parameter space can remain predictively compatible.

The window-level results connect these representation-based measurements to the apparent advantage of chaotic training intervals. Chaotic coverage is associated with lower mean transition-support mismatch ($\rho = -0.60$), and windows with lower support mismatch have lower out-of-support prediction penalties ($\rho = 0.59$). Together, these observations suggest that intervals containing chaotic systems transfer more broadly because they expose the model to a richer set of represented states and transitions. The principal conclusion is therefore that out-of-support transfer is better organized by similarity of the represented process than by proximity in the logistic-map parameter.

A.1 Restricted-window transfer under square loss

We repeat the restricted-window experiment with separately trained models that predict the continuous next state using square loss. The training intervals, input tokenization, trajectory budget, context length, and evaluation grid are unchanged from the cross-entropy experiment. As shown in Figure 7, prediction error remains lowest within the training support and generally increases outside it, although the detailed transfer structure differs between objectives. In particular, the low-loss regions highlighted in the categorical experiment are less apparent under square loss. Nevertheless, represented-state and transition similarity remain associated with out-of-support performance (Table 1), albeit more weakly at the training-window level. Thus, the broad relationship between represented-process coverage and transfer is not specific to cross-entropy, while its finer structure depends on the prediction objective.

Table 1: **Represented-process similarity predicts transfer under both categorical and scalar prediction objectives.** We repeat the restricted-window correlation analysis for separately trained models receiving the same 64-bin inputs: one predicts a 64-bin output with cross-entropy (CE), and the other predicts a scalar with square loss (MSE). For CE, Δ is the same-width, same-target difference in $\log_{10}(1 + \mathrm{CE})$; for MSE it is the corresponding log ratio, $\log_{10}(\mathrm{MSE}/\mathrm{MSE}_{\mathrm{ref}})$. Here D_{state} is context-token occupancy JS divergence, D_1 and D_2 are nearest transition-support distances under Manhattan and Euclidean ground metrics, d_r is distance from the training interval in parameter space, and C_{chaos} is the fraction of the training interval with positive Lyapunov exponent. Pair-level partial correlations control for d_r and categorical window width. Window-level residual correlations remove quadratic window-center and window-width effects. Correlations are descriptive because evaluation targets and training windows overlap.

Association	CE	Scalar MSE
*Window–target level (*14,658 *OOD pairs): Spearman* ρ		
$\rho_{\mathrm{S}}(D_{\mathrm{state}}, \Delta)$	0.560	0.529
$\rho_{\mathrm{S}}(D_1, \Delta)$	0.439	0.485
$\rho_{\mathrm{S}}(D_2, \Delta)$	0.411	0.479
$\rho_{\mathrm{S}}(d_r, \Delta)$	0.073	0.353
Window–target level: partial Spearman ρ		
$\rho_{\mathrm{S}}(D_{\mathrm{state}}, \Delta \mid d_r, w)$	0.570	0.464
$\rho_{\mathrm{S}}(D_1, \Delta \mid d_r, w)$	0.562	0.353
$\rho_{\mathrm{S}}(D_2, \Delta \mid d_r, w)$	0.551	0.345
*Training-window level (*53 *windows): Spearman* ρ		
$\rho_{\mathrm{S}}(C_{\mathrm{chaos}}, \overline{\Delta})$	-0.593	-0.369
$\rho_{\mathrm{S}}(\overline{D}_{\mathrm{state}}, \overline{\Delta})$	0.324	0.328
$\rho_{\mathrm{S}}(\overline{D}_1, \overline{\Delta})$	0.585	0.326
$\rho_{\mathrm{S}}(\overline{D}_2, \overline{\Delta})$	0.531	0.318
Training-window level: residual Pearson r		
$r(C_{\mathrm{chaos}}, \overline{\Delta})$	-0.328	-0.198
$r(\overline{D}_{\mathrm{state}}, \overline{\Delta})$	0.040	0.253
$r(\overline{D}_1, \overline{\Delta})$	0.526	0.322
$r(\overline{D}_2, \overline{\Delta})$	0.532	0.334

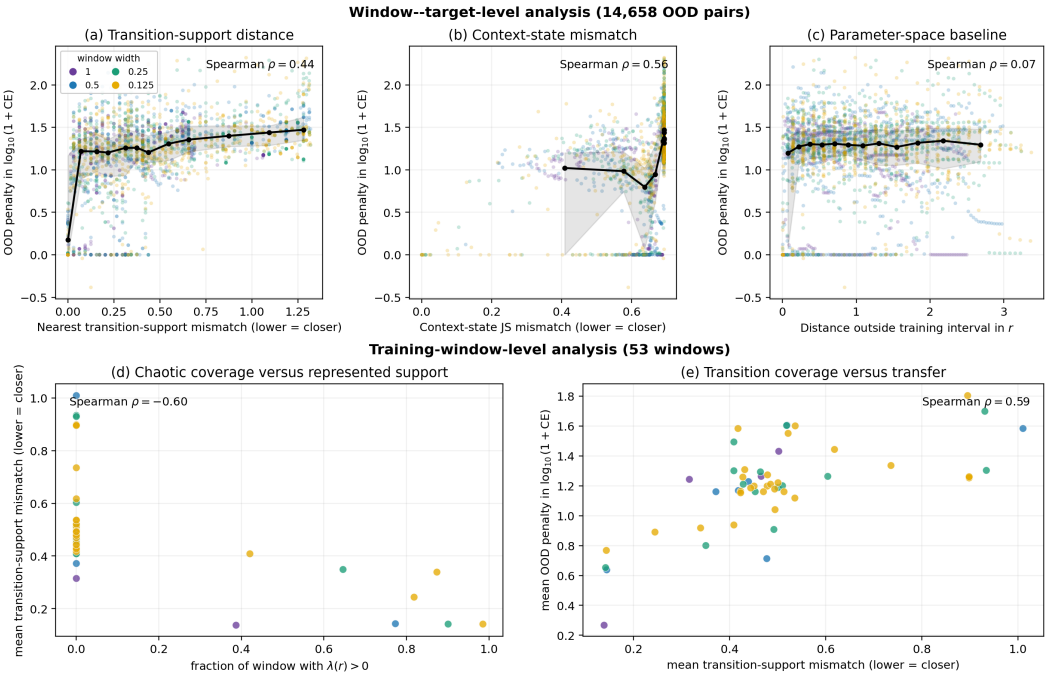

Figure 6: **Represented state and transition similarity are associated with out-of-support transfer.** The top row analyzes 14,658 out-of-support window–target pairs. **(a)** Normalized prediction penalty versus the distance from the target transition mass to the nearest transition support represented within the training interval. **(b)** Penalty versus Jensen–Shannon divergence between the target and training context-state occupancy distributions. **(c)** Penalty versus distance outside the training interval in the logistic-map parameter r. Black curves show medians within predictor quantiles and gray bands show interquartile ranges. The displayed scatter is subsampled for legibility, while correlations and summaries use all pairs. The bottom row averages over targets to give one point per training window. **(d)** Chaotic coverage versus mean transition-support mismatch. **(e)** Mean transition-support mismatch versus mean normalized out-of-support penalty.

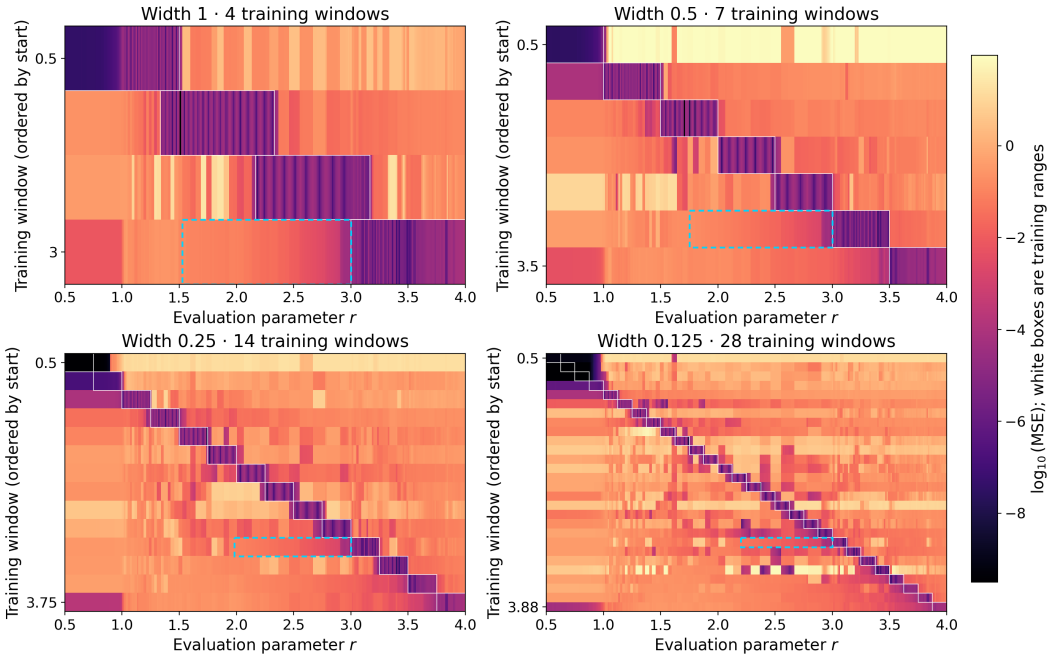

Figure 7: **Restricted-support transfer under scalar next-state prediction.** The experimental protocol matches Figure 4, except that separately trained models predict the continuous next state using square loss rather than a 64-bin output distribution. Rows correspond to training intervals $W = [a, a + w]$, columns to evaluation parameters $r \in [0.5, 4]$, and panels to widths $w \in \{1, 0.5, 0.25, 0.125\}$. Color denotes $\log_{10}(\mathrm{MSE})$ on a shared scale, with darker values indicating lower error. White rectangles mark the training intervals, and cyan dashed rectangles mark the predicted fixed-point/period-two overlap $[r_{\min}(w), 3]$ for models trained on $[3, 3 + w]$. Models receive 8,000 trajectories, 64 input bins, and context length $L = 50$, and are evaluated at fixed 160k-step checkpoints.