PREDICTIVE STATISTICS SHAPE EMERGENT WORLD REPRESENTATIONS OF GRID WALKERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Next-token predictors often appear to develop internal representations of the latent world and its rules. The probabilistic nature of these models suggests a deep connection between the structure of the world and that of sequence probabilities. In order to better understand this link, we use a minimal stochastic process as a controlled setting: constrained random walks on a two-dimensional lattice that must reach a fixed endpoint after a predetermined number of steps. Their world's grid geometry is a statistic for prediction of the full future. We train decoder-only transformers on exactly sampled walks and ask which statistics are decodable and how they are used for computation. The transformer's computation factors into two stages: the first attention block extracts the world statistic from the input by path integration, and later layers transform it into the next-step predictive geometry. Across the fully observed variants the post-attention representation is universal: a shared world-state of the lattice that can be read directly as a world model, which we can trace back to the predictive structure of the data. Later layers then specialize it to each variant's next-step distribution. We also explore a variant with masked steps and latent variables, and the spatial map persists and remains causally used, even though it no longer suffices for prediction. Moreover, we train recurrent networks which solve the fully observed tasks near-optimally, but they organize the computation differently. Although demonstrated in a toy system, these results suggest that world representations emerge as intermediate computations in prediction, with a geometry shaped jointly by the predictive distribution and the architecture that computes it.

1 INTRODUCTION

Deep neural networks trained on sequence prediction often develop internal representations whose geometric organization mirrors structure of the world they are trained on (Li et al., 2023; Gurnee & Tegmark, 2024; Ivanitskiy et al., 2023; Karvonen, 2024; Baumgartner et al., 2025). These observations are broadly suggestive of an emergent "world model" (Vafa et al., 2024; Millière & Buckner, 2024). However, two basic questions remain open: what structure do these representations encode, and why do they take the particular form they do? Mounting evidence suggests that the geometry of internal representations tends to reflect the structure of probability distributions defined over the data. More specifically, recent work hints at the relationship between spatial world models and the predictive distributions of the generating process (Li et al., 2023; Gurnee & Tegmark, 2024; Shai et al., 2024; Baumgartner et al., 2025; Karkada et al., 2026), but it does not directly explore it.

Here, we develop a simple setting to study the relationship between predictive distributions and neural world representations: constrained random walkers on a 2D square lattice. By varying the constraints within the same fixed grid world, we define a tractable family of next-step prediction problems. We use four main variants with incremental degrees of complexity. In three of them, the spatiotemporal variables (x, y, t) are a sufficient statistic for prediction of the full future, and a fourth variant introduces hidden variables and partial observability. These walkers give a flexible setting with a well-defined notion of a spatial world—the square lattice—, controlled latent variables, and an exactly known stochastic law. By training neural networks on sequences of steps, we can precisely ask how emerging world representations relate to the information needed for prediction, the internal structure of the predictive distribution, and the architectural details of the trained network.

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.

Independently trained transformers consistently develop the same embedding across the three fully observed variants and random seeds. This representation is not the straightforward (x, y, t) triplet, but it comes close: we provide empirical evidence that $(x/t, y/t, \log t)$ is a good abstract description of the learned geometry, and trace this representational convergence to a simple attention-based path-integration mechanism. We first show that these coordinates linearly span the post-attention residual stream in a strikingly universal way across independently trained variants and seeds, but not other layers. Moreover, this spatial information is not just present in the network—modeling the underlying lattice geometry is a crucial step in the network's computation of predictions. We find strong causal evidence of this through steering and ablation experiments.

We also show that this representation enables the network to generalize to unseen walks and lattice states, a hallmark of world models. When moves are partly hidden by an unobserved layout of masking cells, so that position no longer suffices for prediction, the same coordinates persist and remain causally used, while the models learn to infer the layout in context at approximately the Bayesian rate. Finally, we train recurrent neural networks (RNNs) on these three tasks and show that they also solve them near-optimally, yet develop a different mechanism without a common separate path integration module. This highlights the dependence of representation and mechanism on architecture.

Although simple, our setting clearly illustrates and reinforces one plausible origin story of world representations in neural sequence models, closely related to the *simplicity bias hypothesis* of Huh et al. (2024): when a statistic for prediction fits well into the network's inductive bias—in this case conditioned by the transformer's self-attention mechanism—, training reliably converges to it.

2 Related work

Emergent world models. Several studies have shown that next-token predictors develop internal representations that function as world models. Early evidence in a controlled setting came from a GPT model trained on Othello move sequences, which developed an emergent internal representation of the board state that could be used to control model outputs via interventions (Li et al., 2023; Nanda et al., 2023b). Similar findings were subsequently reported for chess-playing language models (Karvonen, 2024). In large-scale LLMs, representations have been shown to encode linear structure for space and time (Gurnee & Tegmark, 2024). Complementary evidence from spatial and graph-structured prediction tasks, studied under both standard training and in-context learning, shows that transformers develop structured internal representations of topology, paths, and space (Ivanitskiy et al., 2023; Baumgartner et al., 2025; Park et al., 2025; Park, 2026).

The geometry of predictive representations. Work linking representation geometry to the geometry of probability distributions over the data differs in which distribution it considers. Computational mechanics studies the predictive distribution $\Pr(\text{future} \mid \text{past})$ and its sufficient statistic: Shai et al. (2024) showed that small transformers trained on edge-emitting hidden Markov models develop linear mappings to the belief-state simplex; Piotrowski et al. (2025) subsequently proposed and tested a mechanistic implementation of belief updating in self-attention; and Riechers et al. (2025) extended the analysis to RNNs and a broader class of generative models, all in stationary and ergodic settings. Independently, Zhang et al. (2025) argue from a Bayesian perspective that autoregressive embeddings should encode sufficient statistics for prediction. A separate line studies other distributions: Karkada et al. (2026) show that symmetries of word co-occurrence statistics analytically govern the manifold geometry of trained embeddings. For grids learned in context from random walks, the spatial representation forms shortly before the model's next-token predictions come to respect the grid (Park et al., 2025), and Wurgaft et al. (2026) find, for these grids as for other concepts, that its manifold is approximately isometric to that of the model's next-token distributions, with steering along it moving the predictions accordingly.

Toy models in interpretability. What makes our setting relevant is that it makes it possible to bring the world-model and predictive-geometry perspectives together. Our random walkers, however, are a toy model by design. Toy models have been important in interpretability: they were crucial in the study of superposition (Elhage et al., 2022), the algorithms networks learn for modular arithmetic and group operations (Nanda et al., 2023a; Zhong et al., 2023; Chughtai et al., 2023), the geometry of belief states (Shai et al., 2024), and world representations in board games (Li et al., 2023). The full

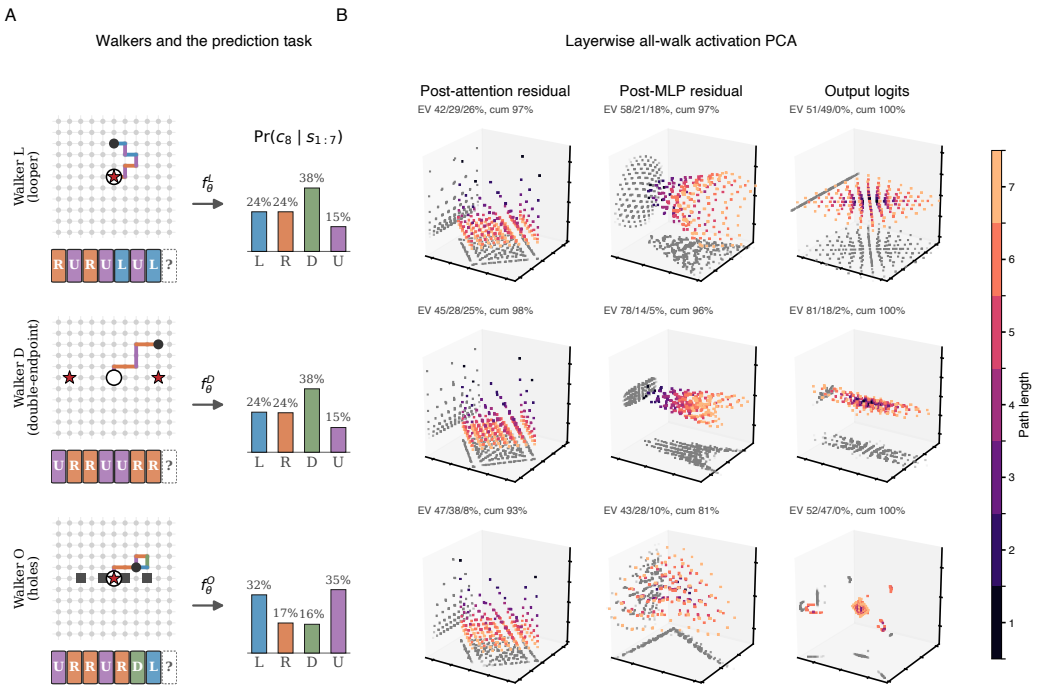

Figure 1: The walkers, the prediction task, and the representation that emerges. Rows: **walker L** (looper), **walker D** (double-endpoint), **walker O** (holes). **(A)** A length-7 prefix on the lattice (one color per move type; start white circle, endpoints red stars, holes dark squares, final position black dot), its token sequence, and the next-step distribution $\Pr(c_8 \mid s_{1:7})$ of the per-walker model $f_\theta^L, f_\theta^D, f_\theta^O$. **(B)** Three-dimensional PCA of the activations of all valid walk prefixes after the first attention block, after the MLP, and at the output logits; each point is one path, colored by length, with principal-component variance fractions annotated. The post-attention geometry is largely shared across walkers, whereas the post-MLP and output geometries are walker-specific: shared structure is built before the walker-specific prediction. Single representative runs at $d_{\text{head}} = 64$ (Walker O: O-2L2H); population statistics in Section 4.1.

object of study is always the whole entangled system of architecture, data and training, and that easily becomes too big to control. Reducing it to a setting where one aspect can be isolated and varied lets us understand that aspect before scaling up, much as biology relies on model organisms (Sharkey et al., 2025).

3 Setup and Methods

3.1 The Grid Walker

We first introduce the family of systems that this work centers on: constrained random walks on a two-dimensional square lattice, which we call *walkers*. The grid's points are labeled by $\boldsymbol{x} = (x, y) \in \mathbb{Z}^2$. The walkers always start from $(0, 0)$ and move for a fixed number of timesteps T. We define four possible displacements, $\Delta_L = (-1, 0)$, $\Delta_R = (1, 0)$, $\Delta_D = (0, -1)$, $\Delta_U = (0, 1)$. Sequences of moves are represented as strings $s_{t_1:t_2} := (c_{t_1}, \ldots, c_{t_2})$, where each move-token c can be L, R, D or U. Moreover, length-t prefixes are abbreviated $s_t := s_{1:t}$, and its corresponding position is the sum of displacements $\boldsymbol{x}_t = \sum_{t'=1}^{t} \Delta_{c_{t'}}$. A close relative of our setting can be found in Jin & Rinard (2024), who probe a language model trained on programs that move a robot on a 2D grid for the robot's unobserved position; there, unlike here, the exact predictive law is not available.

General setting. The walker is parameterized by an endpoint set $\mathcal{P} \subset \mathbb{Z}^2$ and a (possibly empty) set of excluded positions $\mathcal{O} \subset \mathbb{Z}^2$, which we call *holes*. Walks are sampled in two stages: an endpoint

$\boldsymbol{p}$ is randomly picked from $\mathcal{P}$ first, and a length-T path is then sampled among all those ending at $\boldsymbol{p}$ which avoid the holes $\mathcal{O}$. Although full length-T walks are uniformly distributed, our neural networks train on partial walks of length $t < T$. These are not uniform, with probability proportional to the number of valid completions[1] (Appendix A).

Note that the pair $(t, \boldsymbol{x}_t)$ is a sufficient statistic for prediction: conditioned on it, the distribution over future continuations no longer depends on the exact path. Equivalently, $(t, \boldsymbol{x}_t)$ defines a predictive-state, *i.e.*, one which captures sufficient information from the past relevant to predicting the future (Shalizi & Crutchfield, 2001; Littman et al., 2001).

Next-step prediction. The probabilities of the four possible next steps must sum to one, so three independent values suffice to describe the prediction. We thus parameterize the next-step distribution by the *effective field* $\boldsymbol{q}_t = (q_u, q_v, q_{uv})$:

$$\Pr(\Delta \mid s_t; \mathcal{P}, \mathcal{O}) = \frac{\exp(-q_u \Delta u - q_v \Delta v - q_{uv} \Delta u \, \Delta v)}{\sum_{\Delta'} \exp(-q_u \Delta u' - q_v \Delta v' - q_{uv} \Delta u' \, \Delta v')}, \tag{1}$$

where $\Delta u = \Delta x + \Delta y$ and $\Delta v = \Delta x - \Delta y$, and the sum runs over the four possible moves (see Appendix B for more details). Multiple endpoints and holes change the move probabilities by changing the available future paths (Appendix A).

3.2 Data generation

We use three basic walker variants with horizon $T = 20$, which differ in their constraints (Figure 1A):

- **Walker L** (looper): endpoint $\boldsymbol{p} = (0, 0)$, no holes.
- **Walker D** (double-endpoint): endpoints $\mathcal{P} = \{(-4, 0), (4, 0)\}$, no holes.
- **Walker O** (holes): endpoint $\boldsymbol{p} = (0, 0)$, holes at $\{(\pm 1, 0), (\pm 3, 0)\}$.

A fourth, masked variant is introduced in Section 4.5. For these three variants, we sample contexts from the exact prefix marginal, computed by dynamic programming (Equation 4, Appendix A). Unless stated otherwise, the context length is $K = 8$.[2]

3.3 Neural Models

Architectures. We train decoder-only transformers (Vaswani et al., 2017) from scratch on all walkers. An initial architecture sweep varies layer count $L \in \{1, 2\}$, head count $H \in \{1, 2\}$, and head dimension $d_{\text{head}} \in \{32, 64, 128\}$ for Walkers L, D and O (Table 5, appendix). For the main-text figures, we select from this sweep single-layer, single-head models ($L = H = 1$) for Walkers L and D, and three Walker O architectures—one layer with one head, one layer with two heads, and two layers with two heads—abbreviated O-1L1H, O-1L2H and O-2L2H.

Sampling and training. Training samples are drawn in minibatches with replacement from the exact prefix marginal. The objective is next-token cross-entropy, averaged uniformly over all context positions and over the minibatch. Because there is no beginning-of-sequence token, the first prediction is c_2 given c_1. All models are optimized with AdamW; Appendix E.1 gives the training settings.

Validation. For training-distribution performance, we report next-token cross-entropy normalized by the ground-truth conditional entropy at each prediction position, then averaged across positions. Because cross-entropy is bounded below by the conditional entropy, this score is at least 1, with equality for the optimal predictor. For L/D/O, we average cross-entropy over all possible prefixes, weighted by their probabilities, thus measuring fit to the full training distribution. Table 7 in Appendix E.3 reports these scores; generalization tests are described in Section 4.4.

[1] Equivalently, each endpoint-conditioned process is a finite-horizon Doob h-transform of the simple random walk (Doob, 1957; Levin & Peres, 2017).

[2] Shorter prefixes with $t < K$ also contribute to the next-token loss.

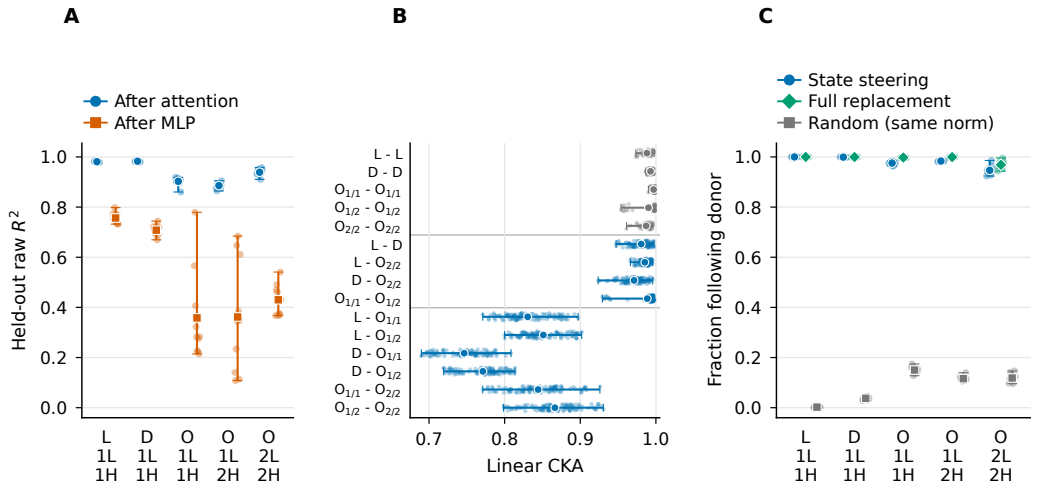

Figure 2: State coordinates span activations, align across models, and support prediction. Five walker–architecture pairs, ten seeds each, at head width 64 and the first transformer block; small points are seeds, large markers and whiskers means and minima–maxima. **(A)** Ten-fold held-out-state R^2 of the affine map from $[x_t/t, y_t/t, \log t]$ to individual activations after attention and after the MLP, with equal total weight per state. **(B)** Pairwise linear CKA between post-attention state-mean representations on each pair's common state support, over all seed pairs: gray rows compare seeds of the same model, followed by the four within-group and six between-group model pairs (Walker O subscripts give layers/heads). **(C)** Fraction of interventions after W_O whose output is closer in forward KL to the donor's exact next-step law than to the original state's, excluding Bayes-equivalent pairs. State steering approaches full activation replacement; norm-matched random steering orthogonal to the state subspace has much less effect. Exact values and details in Appendix D.

4 Experiments and Results

In what follows, we perform several experiments that together reveal how transformers trained on the walkers of Section 3.1 consistently develop neural representations which can be interpreted in two ways at once: informationally, as carriers of the known, tractable predictive statistics; mechanistically, as world models that form the basis of actual network computation. Every transformer in the architecture sweep trains successfully, predicting within 1.5×10^{-4} of the Bayes-optimal normalized loss (Appendix E.3).

4.1 A shared representation precedes task-specific prediction

A world model must separate histories that differ in their future, even when they agree on the next step: Vafa et al. (2024) group histories by the continuations they admit, and Shai et al. (2024), more generally, by the probability distribution they induce over the entire future. In the walkers, the state $(t, \boldsymbol{x}_t)$ fixes this distribution, whereas the effective field $\boldsymbol{q}_t$ (Eq. 1) fixes only the next step. Because the logits are a linear readout of the final LayerNorm (LN-final), and $\boldsymbol{q}_t$ expresses the next-step distribution in logit coordinates, we first test whether $\boldsymbol{q}_t$ can be decoded there. As expected[3], it is decodable on held-out states in all 150 models ($R^2 \geq 0.84$; Figure 10). Immediately after attention, however, the same probe generalizes poorly ($R^2 \leq 0.77$).

The post-attention residual instead carries the state itself, in the form of the *state coordinates* $(x_t/t,\ y_t/t,\ \log t)$, which fix $(t, \boldsymbol{x}_t)$ uniquely and are decodable in all 150 models ($R^2 \geq 0.979$; Figure 10). While seemingly arbitrary, the form of these coordinates becomes clearer from a mechanistic viewpoint in Section 4.3. This representation therefore separates histories that agree on the next step: in Walker O, the positions $(-2, 0)$, $(0, 0)$, and $(2, 0)$ share one next-step distribution,

[3]Although note that Softmax's gauge freedom implies that $\boldsymbol{q}_t$ might have not been linearly decodable.

yet remain apart after attention. Surprisingly, these three features alone also explain most of the post-attention activation variance: fitting the reverse map—from the three features to individual activations after the first attention block—gives held-out R^2 of 0.860–0.986 across all 50 runs at head width 64 (Figure 2A). The lower post-MLP R^2 in the same panel indicates that these features no longer span the activations as faithfully after the nonlinear activations.

The post-attention representations are also remarkably consistent across training runs and walkers. Independently trained seeds of the same model converge to nearly the same representation, with linear CKA (Kornblith et al., 2019) between first-layer post-attention state means of 0.954–1.000, and models trained on L, D, and O-2L2H agree with one another almost as closely (0.924–0.997; Figure 2B). The single-block Walker O models remain aligned with them, if less closely (0.690–0.931); they also carry structure beyond the three coordinates (Figure 2A) and predict slightly worse in every matched-seed comparison (Appendix E.3).

4.2 The representations are used for prediction

Decodability proves that the coordinates are available for computation. However, following the account of representation defended by Geiger et al. (2026), a vehicle represents a property only if the information it carries is also *used* by the system to perform its task. And probes can sometimes decode variables that a model's causal route does not transmit (Darade & Thorat, 2026). To test whether the model computes with this representation, we steer the post-attention residual stream while leaving the input sequence unchanged: using the fitted affine map of Section 4.1, we add the state-coordinate displacement from the original state to a donor state, so that the edit acts only within the three-dimensional state subspace (Appendix D.2). If downstream layers use this map, the output should move toward the donor state's next-step distribution.

Indeed, state-subspace steering makes the output closer in forward KL to the donor's exact law than to the original law in 92.4–100% of interventions (Figure 2C; minima–maxima over 50 runs of per-run averages over donor pairs). This is almost as effective as patching the entire activation space (94.4–100%), whereas random edits of the same norm orthogonal to the state subspace redirect predictions in only 0.0–17.4%.

Ablation results are also compatible with a causally relevant representation: removing the state subspace is at least as damaging as uniform guessing on average for every architecture. We measure forward KL to the target next-step distribution relative to the KL of uniform guessing over the four moves, with equal-rank random ablation as a control (Table 2, Appendix D.2). Random ablation remains better than uniform guessing in all 50 runs, whereas state ablation is worse in $49/50$; it is worse than random ablation in all 50, both here and before the attention output projection W_O. Changing the coordinates redirects the output and removing them disrupts it, so the computation relies on them. We can thus argue that the predictive statistic of Section 4.1 is also a world model.

4.3 Attention constructs an architecture-specific coordinate system

Why does this specific set of state coordinates emerge so consistently at the first self-attention module? The mechanism behind it, common to all walker variants, turns out to be simple and elegant. It combines three properties of the trained attention head (full derivation in Appendix C). Figure 3 shows a representative Walker L run; the same uniform-attention, additive value-write pattern was verified across the trained walker variants and architecture sweep:

1. **Uniform averaging.** The attention head averages value-write contributions approximately uniformly over prefix positions $1, \ldots, t$ (Figure 3A).
2. **Additive value writes.** Each value-write vector decomposes into a token-specific direction code $\boldsymbol{a}_{c_i}$ and a position-specific time code $\boldsymbol{p}_i$, with negligible cross-coupling between the two.
3. **Token and positional codes.** The direction codes arrange the four lattice moves at opposite poles of a 2D subspace (Figure 3B), so their average is a linear function of the mean displacement:

$$\frac{1}{t}\sum_{i=1}^{t}\boldsymbol{a}_{c_i} \;\simeq\; A\frac{1}{t}\sum_{i=1}^{t}\Delta_{c_i} \;=\; A\,\frac{\boldsymbol{x}_t}{t}, \tag{2}$$

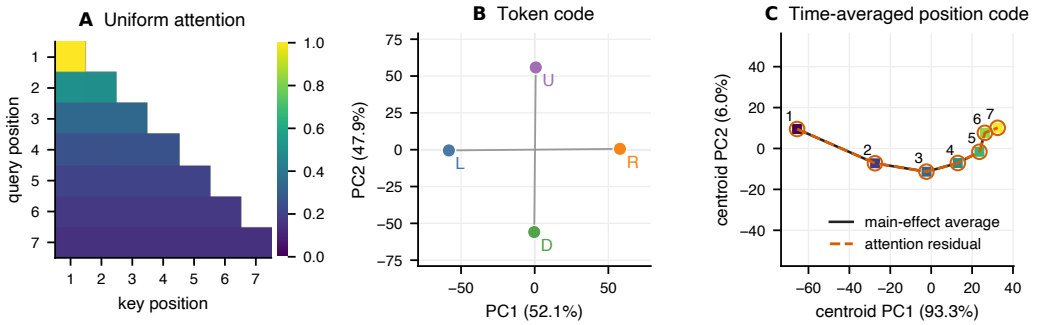

Figure 3: First-layer mechanism in a representative trained Walker L transformer ($L = 1, H = 1, d_{\text{head}} = 128$). **(A)** The single attention head is nearly uniform over the causal prefix. **(B)** The token component of the value-write vectors forms a two-dimensional compass code for the four lattice moves. **(C)** The residual after removing the best affine $x_t/t, y_t/t$ component from the empirical attention output (orange dashed) is almost exactly the time average of the position component of the value-write vectors (black). Panel C is plotted in the PCA basis of the seven empirical residual centroids.

for $A = \begin{bmatrix} \boldsymbol{a}_R^\top \\ \boldsymbol{a}_U^\top \end{bmatrix}$. Averaging the position codes gives the time code $\boldsymbol{h}_t := \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{p}_i$, which depends only on t.

The attention output therefore decomposes into a 2D spatial coordinate $\boldsymbol{x}_t/t$ and a deterministic time code $\boldsymbol{h}_t$; the post-attention panels of Figure 1B show this geometry in the raw activations. As a compact scalar proxy, $\log t$ fits $\boldsymbol{h}_t$ well empirically, though $\boldsymbol{h}_t$ is not strictly one-dimensional (Figure 3C): a second PCA component of the empirical time centroids captures approximately 6% of their variance, so $\log t$ approximates only the dominant direction.

This suggests that the origin of this world representation is a simplicity bias (Huh et al., 2024): the coordinates are what a single head computes most cheaply. Indeed, Bhattamishra et al. (2020) show constructively that uniform attention over value-coded symbols suffices to compute normalized prefix counters, a close analogue of the mechanism derived here.

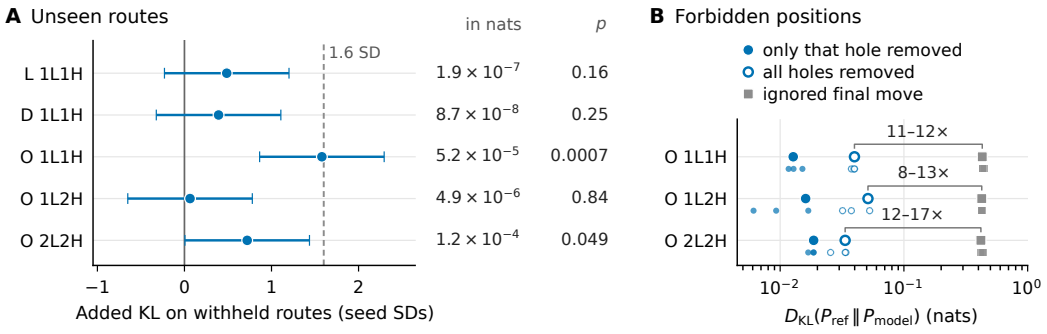

Figure 4: Generalization of representations and predictions. **(A)** Additional prediction error from withholding routes: paired difference-in-differences of forward KL, normalized by its across-seed standard deviation, with 95% paired-t intervals over ten seeds; the dashed line marks 1.6 seed SD, and the columns give the unnormalized mean and the paired-t p-value. **(B)** Forbidden-position predictions compared with three reference laws using $D_{\text{KL}}(P_{\text{ref}} \| P_{\text{model}})$. Large markers pool evaluations over five seeds and three times; small points show time-specific means, and brackets give the range of ignored-move/no-holes KL ratios.

4.4 THE REPRESENTATION GENERALIZES BEYOND THE TRAINING SUPPORT

In the mechanism of Section 4.3, the attention output depends on a prefix only through token averages, and not on the route as a whole. Any prefix built from familiar tokens at familiar positions should therefore be mapped to the coordinates of its endpoint, whether or not that route, or even that endpoint, occurred during training. We test this at two levels: routes omitted from training and positions excluded by the process itself (Figure 4; protocols and additional tests in Appendix D.3).

Unseen routes. Every position remains represented during training, but 10% of the eight-step routes reaching each square are withheld. Notably, withheld routes are assigned to the correct position's activation cluster in over 99.99% of evaluations, indicating that the mechanism of Section 4.3 survives. Moreover, next-step predictions are comparable to those of control models trained on all routes, which shows that withholding routes adds little prediction error (Figure 4A).

Forbidden positions. Walker O contains positions that cannot be occupied at any time. Without retraining, we supply legal prefixes followed by a step onto a hole. The coordinate map fitted on legal inputs decodes every tested post-attention activation to a position nearest the occupied hole's lattice cell. Of three candidate continuation laws, removing only that hole yields the lowest mean forward KL, compared with removing all holes or ignoring the final move (Figure 4B). The map extends into forbidden cells, while the predictions approximate a world in which the occupied hole is absent.

4.5 THE WORLD REPRESENTATION PERSISTS WHEN IT IS NO LONGER SUFFICIENT

In most real settings, knowledge of the world alone is insufficient for prediction (or control), since other latent variables come into play. Moreover, observations are often imperfect and information is partially lost. Does the simple mechanism described in Section 4.3 survive these conditions? We consider an extension, **Walker M** (masked walker), where position and time no longer determine the predictive law by introducing a hidden layout of *masking cells*: leaving a masking cell emits a `MASK` token instead of the direction taken. This hides spatial information from the network, making position uncertain, but does not change how the walker moves. Each sequence concatenates eight loops that return to the origin after 12 steps and share one layout of three masking cells ($K = 96$), so earlier loops inform later predictions, a form of in-context learning that the trained models perform at approximately the Bayesian rate (Appendix D.4). Given the token index, the joint posterior over position and layout is sufficient for prediction, and we compute the next-token distribution exactly by Bayes' rule. To test layout generalization, eight locations never host masking cells during training, and 10% of the layouts avoiding those locations are also withheld (Appendix D.4).

The spatial coordinates found for the previous walkers (Section 4.3) remain visible in Walker M. The mean activation of each (x_t, y_t, t) state follows a clean linear map from $(x_t/t, y_t/t)$ (mean held-out $R^2 = 0.96$), but the same map explains less variance across individual histories ($R^2 = 0.72$). Much of the difference is structured: allowing the baseline activation to depend on loop number and the number of `MASK` tokens observed in that loop raises raw R^2 to 0.89, while also varying the spatial slopes raises it only to 0.91. These gains generalize to held-out states and exceed all 199 matched-label permutations (Appendix D.4), so they cannot be attributed to the additional degrees of freedom alone. The fitted loop- and `MASK`-count shifts lie almost entirely outside the two activation directions encoding x_t/t and y_t/t; smaller slope changes and residual path-dependent variation show that this separation is only approximate. The models still compute with the map: after steering the fitted coordinates toward a donor state, the prediction is closer to the donor's exact law than to its original law in 53–87% of interventions across seeds, compared with at most 1.4% for norm-matched random steering; ablating the fitted subspace leaves the output closer to the donor's law in only 36–55%, near the 50% expected if it favored neither state, compared with at most 1.2% for a random ablation of the same rank. The inferred layout reaches only as far as the training data did: new combinations of familiar masking cells cause no detectable degradation, whereas cells never masked during training are accepted only partly after an early phase in which they were accepted almost fully (Appendix D.4).

4.6 RNNs CARRY AN UPDATEABLE WORLD STATE

Finally, we train 60 LSTM and GRU models ($L \in \{1, 2\}$ layers, hidden sizes $H \in \{16, \ldots, 256\}$) on Walkers L, D, and O; all solve their tasks nearly optimally (Appendix D.6). Since an RNN carries its state from token to token, we examine how that state is updated. A single affine transition $\mathbf{r}_t \approx M\mathbf{r}_{t-1} + \mathbf{b}_{c_t}$, with one matrix M shared by all moves and one offset per observed move token c_t, predicts 99.2–99.6% of the held-out variance of the recurrent state $\mathbf{r}_t$ for Walker L and 96.4–98.8% for Walker D. Under this fit, adding $\mathbf{b}_{\mathrm{R}} - \mathbf{b}_{\mathrm{L}}$ to the recurrent state should shift the walker two cells to the right. It does: from one further update through the end of the context, all 20 networks for each of Walkers L and D predict closer, in forward KL, to the exact law at the shifted position than at the original one, which norm-matched random directions rarely achieve; for Walker O this holds for the first two positions in all 20 networks and less consistently thereafter. The exact effective field $\boldsymbol{q}_t$ also updates almost affinely, so the update rule alone does not show a representation beyond the current prediction, but the recurrent state also separates positions with identical next-move laws, such as $(0, 0)$ and $(\pm 2, 0)$ in Walker O (Appendix D.6). The RNNs thus carry spatial information that they update move by move and that shapes later predictions.

5 CONCLUSION AND LIMITATIONS

Our simple lattice setting makes it possible to connect learned representations to exact predictive distributions and test their role in computation as world models. For the fully observed walkers, the post-attention state coordinates $[x_t/t,\ y_t/t,\ \log t]$ (Sections 4.1–4.2) are at once a sufficient statistic for prediction and a world-state representation of the grid—world coordinates that compactly parameterize the entire future distribution. The grid walkers thus give an analytically tractable instance of the emergent world models found in sequence models trained on games and spatial or graph-structured data (Li et al., 2023; Gurnee & Tegmark, 2024; Ivanitskiy et al., 2023; Karvonen, 2024), and are especially close to map-like representations learned from random walks (Baumgartner et al., 2025; Park et al., 2025)—except that here the map is an explicit predictive statistic. This connects to work casting autoregressive models as encoders of belief states (Shai et al., 2024; Piotrowski et al., 2025; Riechers et al., 2025) or sufficient statistics (Zhang et al., 2025) of the latent process, with the twist that ours is a spatial coordinate system.

The exact form of the world representation is very sensitive to architecture, since each may have its own bias. In transformers, approximately uniform attention over direction codes constructs normalized displacement through averaging, giving a simple route from observed moves to spatial coordinates. This mechanism helps explain why similar maps emerge independently across tasks sharing the same lattice. The recurrent models illustrate another organization, carrying spatial information through successive state updates. Together, these findings suggest that the prediction problem and the available computations jointly shape representation. Thus, the geometry of predictive states alone do not select a unique internal geometry, but strongly condition it.

While walkers shine by the tractability that enables these tests, their scope is naturally restricted. All variants retain simple additive lattice motion, most tokens directly identify moves, and our experiments use small networks and short walks. Moreover, the effect of initialization on our results should be studied further: at four times the default initialization scale, eight of ten runs reach baseline loss yet build a weaker post-attention map, whereas scales from $\times\frac{1}{4}$ to $\times 2$ leave it intact (Appendix E.2). Walker M introduces hidden position and layout, but does not test whether comparable spatial structure emerges when observations reveal much less about movement, or when there is a proliferation of latent variables. More extensively hidden dynamics and richer formal languages, including context-free grammars (Delétang et al., 2023; Strobl et al., 2024), would test how predictive representations organize more complex dependencies. Extending this analysis to larger models and data without tractable predictive laws will also require new ways to identify and intervene on the relevant statistics; decomposing distributions into manageable factors is one promising approach (Shai et al., 2026).

These limitations notwithstanding, we believe there may be stronger constraints on representations, mechanism and world models, which become clearer when the relationship between the underlying probabilistic structure and the architecture is understood. This matters for mechanistic interpretability, where narrowing the space of algorithmic abstractions searched is crucial for feasibility.

AI USE STATEMENT

We used generative AI tools (LLM-based coding and writing assistants) to implement and debug analysis code, to clean and reformat result tables, to help design control and robustness experiments, to help interpret results, and to draft and edit parts of the text. We also used them to search and summarize related literature, to format references, and to check every reported number against the result files it was computed from. We did not use them to generate data (all training data are sampled from exactly computed distributions), to formulate or prove mathematical claims, or to propose the research questions. All AI-assisted code was run and validated against the exact ground-truth distributions, every quoted number was traced to its result file by an author, and all text was reviewed and revised by the authors, who take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

All training and evaluation data are sampled from distributions computed exactly by dynamic programming or Bayesian filtering (Section 3.1, Appendix A), so every model is compared with a ground truth that involves no approximation. Architectures, optimizer settings and training budgets are given in Appendix E.1; probing, intervention and generalization protocols in Appendices D.1–D.3; and every reported number is computed from saved result tables.

REFERENCES

Caroline Baumgartner, Eleanor Spens, Neil Burgess, and Petru Manescu. Cognitive maps in language models: A mechanistic analysis of spatial planning, 2025. URL https://arxiv.org/abs/2511.13371.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, 2020. doi: 10.18653/v1/2020.emnlp-main.576. URL https://aclanthology.org/2020.emnlp-main.576/.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023.

Róbert Csordás, Christopher Potts, Christopher D. Manning, and Atticus Geiger. Recurrent neural networks learn to store and generate sequences using non-linear representations, 2024. URL https://arxiv.org/abs/2408.10920.

Ishita Darade and Sushrut Thorat. Represented is not computed: A causal test of candidate algorithmic intermediates in a transformer, 2026. URL https://arxiv.org/abs/2605.22488.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. Neural networks and the chomsky hierarchy. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WbxHAzkeQcn.

J. L. Doob. Conditional brownian motion and the boundary limits of harmonic functions. *Bulletin de la Société Mathématique de France*, 85:431–458, 1957. URL http://eudml.org/doc/86928.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Atticus Geiger, Jacqueline Harding, and Thomas Icard. How causal abstraction underpins computational explanation, 2026. URL https://arxiv.org/abs/2508.11214.

Wes Gurnee and Max Tegmark. Language models represent space and time. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 2483–2503, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/0a6059857ae5c82ea9726ee9282a7145-Paper-Conference.pdf.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The Platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR, 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Michael Ivanitskiy, Alexander F Spies, Tilman Räuker, Guillaume Corlouer, Christopher Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia Diniz Behn, Katsumi Inoue, and Samy Wu Fung. Linearly structured world representations in maze-solving transformers. In *UniReps: the First Workshop on Unifying Representations in Neural Models*, 2023. URL https://openreview.net/forum?id=pZakRK1QHU.

Charles Jin and Martin Rinard. Emergent representations of program semantics in language models trained on programs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Dhruva Karkada, Daniel J. Korchinski, Andres Nava, Matthieu Wyart, and Yasaman Bahri. Symmetry in language statistics shapes the geometry of model representations, 2026. URL https://arxiv.org/abs/2602.15029.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=9n2s8l7XoQ.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. doi: 10.48550/ARXIV.1905.00414.

David A. Levin and Yuval Peres. *Markov Chains and Mixing Times*. American Mathematical Society, Providence, RI, 2 edition, 2017. URL https://pages.uoregon.edu/dlevin/MARKOV/. With contributions by Elizabeth L. Wilmer.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *International Conference on Learning Representations*, 2023.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1555–1561, 2001. URL https://papers.neurips.cc/paper/1983-predictive-representations-of-state.

Aleksandar Makelov, Georg Lange, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching, 2023. URL https://arxiv.org/abs/2311.17030.

Raphaël Millière and Cameron Buckner. A philosophical introduction to language models – part II: The way forward, 2024. URL https://arxiv.org/abs/2405.03207.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023a.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models, 2023b. URL https://arxiv.org/abs/2309.00941.

Core Francisco Park. Convergent world representations and divergent tasks. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Z4PjCncxzz.

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. ICLR: In-context learning of representations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pXlmOmlHJZ.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2024. URL https://arxiv.org/abs/2311.03658.

Mateusz Piotrowski, Paul M. Riechers, Daniel Filan, and Adam S. Shai. Constrained belief updates explain geometric structures in transformer representations. In *Forty-second International Conference on Machine Learning*, 2025. doi: 10.48550/ARXIV.2502.01954.

Paul M. Riechers, Thomas J. Elliott, and Adam S. Shai. Neural networks leverage nominally quantum and post-quantum representations, 2025.

Adam Shai, Loren Amdahl-Culleton, Casper L. Christensen, Henry R. Bigelow, Fernando E. Rosas, Alexander B. Boyd, Eric A. Alt, Kyle J. Ray, and Paul M. Riechers. Transformers learn factored representations, 2026.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers represent belief state geometry in their residual stream. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. doi: 10.48550/ARXIV.2405.15943. URL https://openreview.net/forum?id=YIB7REL8UC.

Cosma Rohilla Shalizi and James P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104(3):817–879, 2001. ISSN 1572-9613. doi: 10.1023/A:1010388907793.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024. doi: 10.1162/tacl_a_00663. URL https://aclanthology.org/2024.tacl-1.30/.

Hugo Touchette. The large deviation approach to statistical mechanics. *Physics Reports*, 478(1):1–69, 2009. ISSN 0370-1573. doi: 10.1016/j.physrep.2009.05.002.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=aVK4JFpegy.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Daniel Wurgaft, Can Rager, Matthew Kowal, Vasudev Shyam, Sheridan Feucht, Usha Bhalla, Tal Haklay, Eric Bigelow, Raphael Sarfati, Thomas McGrath, Owen Lewis, Jack Merullo, Noah Goodman, Thomas Fel, Atticus Geiger, and Ekdeep Singh Lubana. Manifold steering reveals the shared geometry of neural network representation and behavior, 2026. URL https://arxiv.org/abs/2605.05115.

Liyi Zhang, Michael Y. Li, R. Thomas McCoy, Theodore Sumers, Jian-Qiao Zhu, and Thomas L. Griffiths. What should embeddings embed? autoregressive models represent latent generating distributions. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=YyMACp98Kz. Featured Certification.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Computation of Probability Distributions

All probability distributions used for training and validation are computed exactly via a dynamic-programming recursion over the path-count propagator, with no sampling. We follow the notation of Section 3.1: walkers are parameterized by an endpoint set $\mathcal{P} \subset \mathbb{Z}^2$ and hole set $\mathcal{O} \subset \mathbb{Z}^2$ (possibly empty), with $\tau = T - t$ the remaining number of steps after a length-t prefix.

Path-count propagator. For an endpoint $\boldsymbol{p} \in \mathcal{P}$, let

$$H^{\boldsymbol{p}}_{\mathcal{O}}(\tau, z) \;:=\; \#\{\text{length-}\tau \text{ paths from } z \text{ to } \boldsymbol{p} \text{ avoiding } \mathcal{O}\}.$$

This propagator counts valid futures from a given position and remaining time. It satisfies the linear four-neighbor recursion

$$H^{\boldsymbol{p}}_{\mathcal{O}}(\tau+1, z) \;=\; \mathbf{1}_{z \notin \mathcal{O}} \sum_{\Delta \in \{\Delta_L, \Delta_R, \Delta_D, \Delta_U\}} H^{\boldsymbol{p}}_{\mathcal{O}}(\tau, z+\Delta), \tag{3}$$

with terminal condition $H^{\boldsymbol{p}}_{\mathcal{O}}(0, z) = \mathbf{1}_{z=\boldsymbol{p}}$. In the hole-free case ($\mathcal{O} = \emptyset$), the diagonal coordinates $u_z := x_z + y_z$, $v_z := x_z - y_z$ (and $p_u := p_x + p_y$, $p_v := p_x - p_y$ for the endpoint) decouple the recursion and the propagator factorizes in closed form,

$$H^{\boldsymbol{p}}_{\emptyset}(\tau, z) \;=\; \binom{\tau}{(\tau + p_u - u_z)/2} \binom{\tau}{(\tau + p_v - v_z)/2},$$

nonzero only when both binomial arguments are integers in $[0, \tau]$. With holes, we iterate the recursion from $\tau = 0$ to $\tau = T$ and cache every level.

Prefix distribution. Conditioned on the endpoint $\boldsymbol{p}$, all valid length-T paths from origin to $\boldsymbol{p}$ avoiding $\mathcal{O}$ are equiprobable, so the marginal probability of a length-t prefix s_t ending at position $\boldsymbol{x}_t$ is the fraction of full paths that begin with s_t:

$$\Pr(s_t \mid \boldsymbol{p}) \;=\; \frac{H^{\boldsymbol{p}}_{\mathcal{O}}(\tau, \boldsymbol{x}_t)}{H^{\boldsymbol{p}}_{\mathcal{O}}(T, \mathbf{0})}. \tag{4}$$

This is proportional to the number of valid completions from $\boldsymbol{x}_t$ to $\boldsymbol{p}$, and is therefore not uniform in s_t.

Endpoint posterior. For multi-endpoint walkers, the posterior over endpoints given the prefix follows from Bayes' rule with a uniform endpoint prior,

$$w_t(\boldsymbol{p} \mid \boldsymbol{x}_t) \;=\; \frac{\Pr(s_t \mid \boldsymbol{p})}{\sum_{\boldsymbol{p}' \in \mathcal{P}} \Pr(s_t \mid \boldsymbol{p}')},$$

and the unconditional prefix marginal is $\Pr(s_t) = |\mathcal{P}|^{-1} \sum_{\boldsymbol{p}} \Pr(s_t \mid \boldsymbol{p})$.

Next-step probabilities. By Equation 3, the conditional next-step law given endpoint $\boldsymbol{p}$ is the finite-horizon Doob h-transform, with $H^{\boldsymbol{p}}_{\mathcal{O}}$ playing the space-time harmonic function (Doob, 1957; Levin & Peres, 2017):

$$\Pr(\Delta \mid s_t; \boldsymbol{p}) \;=\; \frac{H^{\boldsymbol{p}}_{\mathcal{O}}(\tau - 1, \boldsymbol{x}_t + \Delta)}{H^{\boldsymbol{p}}_{\mathcal{O}}(\tau, \boldsymbol{x}_t)}, \tag{5}$$

which automatically vanishes when $\boldsymbol{x}_t + \Delta \in \mathcal{O}$ and is well-defined because the four numerator terms sum to the denominator. The multi-endpoint distribution is the posterior mixture

$$\Pr(\Delta \mid s_t; \mathcal{P}, \mathcal{O}) = \sum_{\boldsymbol{p} \in \mathcal{P}} w_t(\boldsymbol{p} \mid \boldsymbol{x}_t)\, \Pr(\Delta \mid s_t; \boldsymbol{p}, \mathcal{O}), \tag{6}$$

where the hole set is fixed in both the endpoint posterior and the endpoint-conditioned laws.

Implementation. The recursion is run in diagonal coordinates, where the four-neighbor update becomes four corner additions on a $(T+1) \times (T+1)$ array and the parity constraint $u_z + v_z \equiv \tau \pmod 2$ is enforced automatically. All arithmetic is performed in log-space. Next-step tables for every equivalence class $(\boldsymbol{x}_t, t)$ are computed once via Equation 5 and cached as training batches and validation targets.

B Predictive statistics and the effective field

This appendix derives the three-coordinate description of the next-step law used in Section 3.1, then gives closed-form expressions for the walkers without holes.

Probability and logit coordinates. In diagonal coordinates $u_t = x_t + y_t$ and $v_t = x_t - y_t$, the four lattice moves correspond to the four pairs $(\Delta u, \Delta v) \in \{\pm 1\}^2$. The functions $\{1, \Delta u, \Delta v, \Delta u\, \Delta v\}$ form a complete orthogonal basis on these four outcomes, so every next-step distribution admits the exact decomposition

$$\Pr(\Delta \mid s_t; \mathcal{P}, \mathcal{O}) = \tfrac{1}{4}\big(1 + \langle \Delta u \rangle\, \Delta u + \langle \Delta v \rangle\, \Delta v + \langle \Delta u\, \Delta v \rangle\, \Delta u\, \Delta v\big), \tag{7}$$

where expectations are taken under the next-step law and Δ ranges over the four moves. For strictly positive probabilities, write $\ell_\Delta = \log \Pr(\Delta \mid s_t; \mathcal{P}, \mathcal{O})$ and $\bar{\ell} = \frac{1}{4}\sum_\Delta \ell_\Delta$. The same basis expresses the centered log-probabilities as

$$\ell_\Delta - \bar{\ell} = -q_u\, \Delta u - q_v\, \Delta v - q_{uv}\, \Delta u\, \Delta v, \tag{8}$$

which defines the effective field $\boldsymbol{q}_t = (q_u, q_v, q_{uv})$. Exponentiating and normalizing gives

$$\Pr(\Delta \mid s_t; \mathcal{P}, \mathcal{O}) = \frac{e^{-\boldsymbol{q}_t \cdot \boldsymbol{\Delta}_\star}}{\sum_{\Delta'} e^{-\boldsymbol{q}_t \cdot \boldsymbol{\Delta}'_\star}}, \qquad \boldsymbol{\Delta}_\star = (\Delta u, \Delta v, \Delta u\, \Delta v). \tag{9}$$

These decompositions do not assume independence between the diagonal steps and therefore also apply to walkers with multiple endpoints or holes. Zero-probability moves are obtained as limits of the softmax expression; their exact log-probabilities and the corresponding field need not be finite.

Walker L: closed-form artanh. Without holes and conditioned on a single endpoint $\boldsymbol{p}$, the two diagonal coordinates follow independent one-dimensional bridges. Each must cover its remaining displacement in $\tau = T - t$ steps, giving mean increments $\mu_p = (p_u - u_t)/\tau$ and $\nu_p = (p_v - v_t)/\tau$, where $p_u = p_x + p_y$ and $p_v = p_x - p_y$. The next-step distribution therefore factorizes:

$$\Pr(\Delta \mid s_t; \boldsymbol{p}) \;=\; \tfrac{1}{4}\big(1 + \mu_p\, \Delta u\big)\big(1 + \nu_p\, \Delta v\big), \tag{10}$$

Taking logarithms and applying the identity $\log(1 \pm z) = \frac{1}{2}\log(1 - z^2) \pm \operatorname{artanh}(z)$ separately to the two factors,

$$\log \Pr(\Delta \mid s_t; \boldsymbol{p}) \;=\; \operatorname{artanh}(\mu_p)\, \Delta u + \operatorname{artanh}(\nu_p)\, \Delta v + C(\mu_p, \nu_p),$$

with offset $C(\mu_p, \nu_p) = \frac{1}{2}\log[(1 - \mu_p^2)(1 - \nu_p^2)] - \log 4$ independent of Δ. Matching against Equation 8 reads off the three components of the effective field,

$$q_u \;=\; -\operatorname{artanh}(\mu_p), \qquad q_v \;=\; -\operatorname{artanh}(\nu_p), \qquad q_{uv} \;=\; 0, \tag{11}$$

confirming that Walker L's effective field is effectively 2D. In the language of large deviations, q_u and q_v are the tilting parameters conjugate to the required drift, so Equation 9 is the tilted, exponential-family form of the unconstrained step distribution (Touchette, 2009). The artanh logits diverge at the admissible-domain boundary $|\mu_p| = 1$ or $|\nu_p| = 1$, where one or more next steps become impossible and the Softmax form of Equation 9 is the appropriate limit.

Walker D: bridge-field interpretation. For two endpoints $\mathcal{P} = \{\boldsymbol{p}_+, \boldsymbol{p}_-\}$ with posterior weights $w_\pm = w_t(\boldsymbol{p}_\pm \mid \boldsymbol{x}_t)$ ($w_+ + w_- = 1$), the next-step distribution is the posterior mixture (Equation 6). Linearity of expectation through the product form of Equation 10 immediately gives the next-step moments as posterior averages of the per-endpoint bridge fields,

$$\langle \Delta u \rangle = \langle \mu \rangle_{\mathcal{P}}, \qquad \langle \Delta v \rangle = \langle \nu \rangle_{\mathcal{P}}, \qquad \langle \Delta u\, \Delta v \rangle = \langle \mu\nu \rangle_{\mathcal{P}},$$

where $\langle f \rangle_{\mathcal{P}} := \sum_{\boldsymbol{p}} w_t(\boldsymbol{p} \mid \boldsymbol{x}_t)\, f(\boldsymbol{p})$ and μ, ν are the per-endpoint drifts of Equation 10. Writing $s := w_+ - w_- \in [-1, 1]$ for the belief asymmetry, the third moment satisfies the covariance identity

$$\langle \mu\nu \rangle_{\mathcal{P}} - \langle \mu \rangle_{\mathcal{P}} \langle \nu \rangle_{\mathcal{P}} \;=\; \tfrac{1}{4}(1 - s^2)\,(\mu_+ - \mu_-)(\nu_+ - \nu_-), \tag{12}$$

which is nonzero whenever both endpoints retain posterior support ($|s| < 1$) and their bridge drifts differ in both components. For the symmetric endpoints $\boldsymbol{p}_\pm = (\pm 4, 0)$ of Walker D used in our experiments, $\mu_+ - \mu_- = \nu_+ - \nu_- = 8/\tau$ and Equation 12 specializes to $16(1 - s^2)/\tau^2$. The product factorization of Equation 10 is therefore broken at every prefix where the endpoint posterior is non-degenerate, so $q_{uv} \neq 0$ in the effective-field decomposition (Equation 8) and all three components of $\boldsymbol{q}_t$ carry independent information about position and belief.

C First-Layer Attention Mechanism

We derive the coordinates produced by the first attention layer (Section 4.3). Following the transformer-circuits decomposition of attention heads into attention-routing and value-writing components (Elhage et al., 2021), for a source position i and token c_i, the value-write vector is $w(c_i, i) := v(c_i, i)W_O$, where $v(c_i, i)$ is the head's value vector and W_O is the output matrix. Although the value vector is computed after LayerNorm and could in principle couple token and position nonlinearly, empirically we observe an additive main-effects decomposition:

$$w(c_i, i) \;\approx\; b + a_{c_i} + p_i, \tag{13}$$

where a_{c_i} is a token code and p_i is a position code. The token–position interaction accounts for less than 2×10^{-5} of the centered value-write variance in the Walker L model shown in Figure 3.

The attention pattern is nearly uniform over the causal prefix: at query position t, the head approximately averages values from positions $1, \ldots, t$. Combining uniform averaging with Equation 13 gives

$$\mathbf{o}_t^{\text{attn}} \;\approx\; b + \frac{1}{t}\sum_{i=1}^{t} a_{c_i} + \frac{1}{t}\sum_{i=1}^{t} p_i. \tag{14}$$

The token-average term equals $A\,\boldsymbol{x}_t/t$ for some linear map A (Equation 2). The position-average term,

$$h_t := \frac{1}{t}\sum_{i=1}^{t} p_i, \tag{15}$$

is path-independent and depends only on t. Figure 3C verifies the decomposition: after removing the best-fit affine $x_t/t, y_t/t$ component from the empirical attention output, the residual time centroids coincide with h_t.

D Representational Analysis

D.1 Metrics

We characterize the alignment between an activation $\boldsymbol{a}_\ell(s_t) \in \mathbb{R}^{d_\ell}$ at hook ℓ and a target vector $\boldsymbol{y}(s_t) \in \mathbb{R}^d$ (typically either the state coordinates $\boldsymbol{z}_t = [x_t/t,\, y_t/t,\, \log t]$ or the effective field $\boldsymbol{q}_t$, see Appendix B) using two complementary tools. Both restrict to linear structure in the activations, in line with the empirical observation that task-relevant features in trained networks tend to be encoded as linear directions on which downstream components act (Park et al., 2024). Note, however, there are exceptions to this empirical regularity. For instance, in gated recurrent networks trained to repeat a token sequence, the token at each position has been found to be encoded by activation magnitude rather than by a direction (Csordás et al., 2024).

Affine probe and R^2. We fit an affine map $\boldsymbol{a} \mapsto W\boldsymbol{a} + \boldsymbol{b}$ (or its converse $\boldsymbol{y} \mapsto W'\boldsymbol{y} + \boldsymbol{b}'$) by least squares (with the weights specified below) and report the joint R^2 across all output dimensions. The two probe directions are not redundant and answer different questions:

- **Decoding** ($\boldsymbol{a} \to \boldsymbol{y}$). How recoverable is the target from a linear function of the activation? A high decoding R^2 means $\boldsymbol{y}$ lies (up to noise) in the column space of $\boldsymbol{a}$; it does not constrain what else $\boldsymbol{a}$ may contain.
- **Encoding** ($\boldsymbol{y} \to \boldsymbol{a}$). How much of the activation variance is contained in the affine image of the target? Provided $d \ll d_\ell$, a high encoding R^2 constrains $\boldsymbol{a}$ to mostly lie in a low-dimensional affine subspace spanned by $\boldsymbol{y}$.

Both directions appear in this paper; the relevant text and figure captions state which direction (and which aggregation and which evaluation, see below) is plotted in each panel.

Individual activations and state weighting. For Walkers L, D, and O, prefixes reaching the same position at the same time share the same targets $\boldsymbol{z}_t$ and $\boldsymbol{q}_t$. For the $K = 8$ analyses, encoding fits predict individual activations, weighting each observation by the reciprocal of its state's count so

Table 1: Encoding individual activations after the first attention block, at head width 64. Cells give means over ten seeds, with minima–maxima beneath. Fits use ten held-out-state folds and equal total weight per state. Within-state variance is expressed as a percentage of total variance about the globally weighted mean.

Model	Encoding R^2	Within-state variance (%)
L (1L1H)	0.9809 [0.9779–0.9858]	0.206 [0.171–0.256]
D (1L1H)	0.9828 [0.9789–0.9865]	0.218 [0.184–0.249]
O (1L1H)	0.9027 [0.8602–0.9184]	0.830 [0.588–1.278]
O (1L2H)	0.8858 [0.8644–0.9051]	0.404 [0.337–0.525]
O (2L2H)	0.9391 [0.9103–0.9575]	0.111 [0.077–0.150]

that every state has equal total weight. The score includes variation between histories reaching the same state as unexplained error. In cross-validation, all observations of a state remain in the same fold, and the baseline is the weighted mean activation of the training observations. The $K = 8$ analyses enumerate seven-token histories, retain valid histories, and evaluate positions $t = 1, \ldots, 7$; the architecture sweep uses the common valid support across walkers.

Table 1 reports individual-activation encoding scores for the fifty models in Figure 2A. Within-state variation accounts for 0.077%–1.278% of total post-attention activation variance under these weights, and at most 0.269% after the MLP.

For the **decoding** direction (activations $\rightarrow$ target) we use raw per-position datapoints, to be faithful to the distribution the network was actually trained on. The network processes individual sequences, not state averages, so asking whether the target is recoverable from a single-path activation is the natural question. Although $n_{\text{samples}} \gg d_{\text{source}}$, transformer activations take only about 200 distinct state targets while d_{source} reaches 256, so an in-sample probe can nearly interpolate them; transformer decoding scores are therefore held out over whole states, with ridge regularization chosen by nested cross-validation. Decoding uses equal weight per observation; encoding uses equal total weight per state. The interventions in Figure 2C and Table 2 act on individual sequence activations.

In-sample vs. held-out evaluation. All probes reported are held out with 10-fold cross-validation. Folds partition whole states for encoding and for transformer decoding (Figure 10), and individual observations for RNN decoding (Figure 15). For transformer decoding, folds are also stratified by time step t, so every training fold contains every position: Walker O has only two states at $t = 1$, and a fold holding out both would test extrapolation to an unseen position embedding rather than decodability. The ridge penalty is chosen for each outer fold by five-fold state-grouped cross-validation on its training states. Figure 2A aggregates ten independently trained seeds per walker–architecture pair; a seed changes both initialization and the training data stream. Figure 10 likewise uses ten seeds at each of three head widths. Captions state the evaluation and population used.

Linear centered kernel alignment (lCKA). We measure run-to-run representational similarity with linear CKA (Kornblith et al., 2019), computed on (mean-centered) state-mean activations of two runs r, r':

$$\mathrm{lCKA}\big(\bar{A}_r, \bar{A}_{r'}\big) = \frac{\|\bar{A}_r^\top \bar{A}_{r'}\|_F^2}{\|\bar{A}_r^\top \bar{A}_r\|_F \, \|\bar{A}_{r'}^\top \bar{A}_{r'}\|_F},$$

with $\bar{A}_r \in \mathbb{R}^{n_{\text{states}} \times d_r}$. All lCKA values reported in the paper are computed on state-mean activations. For the ten-seed cross-variant comparisons in Table 6, each model pair is restricted to its pairwise intersection of valid prefixes, and activations are averaged by shared $(\boldsymbol{x}_t, t)$ state class. Models are compared at matched seeds at the first post-attention residual. The earlier sweep diagnostics in Figure 11 instead use the global support intersection. Each CKA value compares two representations of the same ordered set of state classes.

R^2 **vs. lCKA.** The key difference between the two metrics is that R^2 ignores the variance structure of the source space (it maps from sources to targets regardless of which directions carry variance), whereas lCKA down-weights directions of low variance in the source space, thus erasing the distinction between source and target.

D.2 Intervention protocol and additional results

We evaluate five walker–architecture pairs at head width 64 (L-1L1H, D-1L1H, O-1L1H, O-1L2H, O-2L2H), with ten independently trained seeds each. At each intervention site we fit an affine map $\boldsymbol{a} \simeq W_z \boldsymbol{z} + \boldsymbol{b}$ from the state coordinates $\boldsymbol{z} = [x/t, y/t, \log t]$ to state-mean activations. The fit defines the three-dimensional state subspace. Let P_z be the orthogonal projector onto the column space of W_z. For an original state o and donor state d, state steering changes one sequence activation to $\boldsymbol{a} + W_z(\boldsymbol{z}_d - \boldsymbol{z}_o)$, while state ablation changes it to $(I - P_z)\boldsymbol{a}$. Full activation replacement uses an actual donor sequence's activation. Random steering uses an orthogonal displacement with the same norm as the state-steering edit; random ablation projects out a random subspace of rank three. All other sequence positions remain unchanged.

Recipients are states at $t = 6$, and the edit is applied at that position. Donors include different positions at the same time, states at different times, and clock-only pairs with identical x/t and y/t. Pairs are drawn uniformly over states rather than under the walker's own visitation measure, so rare states weigh as much as common ones: 600 same-time and 600 cross-time pairs are sampled at random, draws that land on a Bayes-equivalent pair are discarded rather than redrawn, and every clock-only pair is enumerated and entered ten times, each with its own recipient walk. This gives 1,640 (L), 1,660 (D), and 1,497 (O) donor pairs per run. The pair set is generated once from a fixed random seed and shared by all ten training seeds of a walker (and by the three O architectures), so variation across seeds reflects the trained models only. We exclude Bayes-equivalent donor pairs from the reported intervention metrics. For an intervened prediction p_{edit} and the exact next-step laws p_o, p_d, an intervention follows the donor when $D_{\mathrm{KL}}(p_d \| p_{\mathrm{edit}}) < D_{\mathrm{KL}}(p_o \| p_{\mathrm{edit}})$. We average this indicator within each run and report variation across seeds. Full replacement averages over the sampled donor routes as well as state pairs. For ablation, Table 2 reports mean absolute forward KL, $D_{\mathrm{KL}}(p_o \| p_{\mathrm{ablated}})$, alongside the unedited baseline and $D_{\mathrm{KL}}(p_o \| u)$ for $u = (1/4, 1/4, 1/4, 1/4)$. All conditions use the same recipient population and weights, including repeated states. The uniform reference is therefore $\log 4 - \mathbb{E}_o[H(p_o)]$; it assigns equal probability to all four moves, including blocked moves. These values measure prediction error, not uncertainty between the original and donor laws.

Figure 2C and Table 2 intervene on the first post-attention residual, after W_O. Table 3 repeats the comparisons on the concatenated head outputs before W_O, refitting the state subspace at that site; its ablation columns report paired increases in KL relative to the unedited model. At both sites, state ablation increases KL more than random ablation in all 50 runs. The primary site is the residual stream, which Makelov et al. (2023) recommend for subspace interventions, as against MLP hidden activations, where the high-dimensional kernel of the down-projection supplies a causally disconnected component and an intervened subspace can appear to mediate a feature it does not carry. They pair that recommendation with validations beyond end-to-end causal effect; here those are the encoding R^2 at the same site (Appendix D.1) and the explicit first-layer mechanism of Appendix C. The state subspace is moreover fitted by least squares against the analytic state coordinates rather than optimized for causal effect, so it is not produced by the search procedure for which that illusion was demonstrated.

D.3 Generalization protocols

This appendix specifies the four evaluations of Section 4.4.

Held-out contexts. For each walker–architecture pair we train two arms at context length 16 with paired seeds 1–10 (a seed fixes initialization and data order jointly): one on the full process and one with a fixed, stratified tenth of the enumerated length-8 contexts reaching each terminal cell removed from the training support (6,558 contexts for Walkers L and D, 1,802 for Walker O; the removed set is chosen once and does not vary with the training seed). Both arms are evaluated on exactly the removed contexts and on a per-terminal-cell matched sample of retained contexts, by

Table 2: Post-attention ablation error relative to uniform guessing. Cells report $D_{\mathrm{KL}}(p_o \| p_{\mathrm{ablated}})/D_{\mathrm{KL}}(p_o \| u)$, where p_o is the original state's exact law and u is uniform over the four moves; 1 therefore denotes uniform-guessing error, and lower is better. Values are means over ten seeds; brackets give seed minima–maxima. All values are rounded to three decimal places and use matched recipient states and weights.

Model	Random rank 3	State
L (1L1H)	0.009 [0.001–0.044]	1.026 [0.999–1.161]
D (1L1H)	0.010 [0.004–0.024]	1.194 [1.074–1.317]
O (1L1H)	0.168 [0.061–0.463]	2.958 [1.185–6.275]
O (1L2H)	0.064 [0.019–0.192]	5.184 [2.914–8.898]
O (2L2H)	0.027 [0.005–0.120]	6.416 [1.080–15.722]

Table 3: Interventions before the first attention output projection W_O, at head width 64. Entries are means over ten seeds. Steering uses the same donor-following criterion as Figure 2C; random steering is norm-matched and orthogonal to the state subspace. Ablation entries report the increase in forward KL to the original state's exact law relative to the unedited model, paired within each run; random ablation removes a subspace of the same rank. Bayes-equivalent donor pairs are excluded.

	Fraction following donor			Ablation ΔKL (nats)	
Model	State	Full patch	Random	State	Random
L (1L1H)	1.000	1.000	0.001	0.100	0.00068
D (1L1H)	0.998	0.999	0.012	0.066	0.00043
O (1L1H)	0.973	0.990	0.066	0.814	0.01535
O (1L2H)	0.983	0.997	0.052	0.855	0.00301
O (2L2H)	0.946	0.968	0.064	2.036	0.00190

forward KL from the exact law at the final context position. Because the removed set is chosen by a stratified rule, even the unrestricted arm scores slightly worse on it, so the reported effect is the difference-in-differences: (removed minus retained) in the censored arm minus the same gap in the identically seeded unrestricted arm, averaged over the ten seeds with a paired t-test. We report differences in nats rather than ratios because forward KL is additive over contexts and a ratio of two near-zero divergences is unstable. Representational integrity is a nearest-centroid classification of each removed-context activation, over the full first-layer post-attention residual vector at the final position, to the terminal-cell centroids fitted on retained contexts, against the leave-one-out accuracy of the retained contexts; the reported accuracies are over the 50 censored runs. Mean additional forward KL is 1.9×10^{-7}, 8.7×10^{-8}, 5.2×10^{-5}, 4.9×10^{-6}, and 1.2×10^{-4} nats for L, D, O-1L1H, O-1L2H, and O-2L2H, respectively. All five means are positive and below 1.6 standard deviations of the paired seed differences; the O-1L1H and O-2L2H effects have unadjusted paired-t p-values of 0.0007 and 0.049, so the result is a small penalty rather than evidence of no effect. Only 2 of $185{,}220$ removed-route activations are assigned to the wrong state across the 50 censored runs. Censoring does leave a trace at short prefixes: averaged over all context positions rather than the final one, the censored two-block Walker O models show elevated KL on removed and retained contexts alike, a property of the censored training distribution that the final-position measure separates from the question of whether the removed contexts themselves are mispredicted.

Held-out states. Two lattice states, $(2, 0)$ and $(1, 1)$ at $t = 4$, are given zero measure in Walker L's training process (horizon $T = 20$, endpoint $(0, 0)$, $K = 8$); one- and two-layer models are trained on the resulting process with ten seeds each. Evaluation draws $4{,}096$ sequences from the unrestricted law that pass through a removed state at $t = 4$ and reports the probability the model assigns to the entering move, against the exact unrestricted predictor, and the cross-entropy over the positions after arrival, against the exact unrestricted predictor on the same positions. The models assign the entering move mean probability 0.000102–0.000210 across runs, compared with 0.259146 under the unrestricted law. Cross-entropy up to and including entry is 3.91–4.19 nats against 1.3686 for the reference, while post-arrival cross-entropy is 1.365291–1.365409 against 1.365330. These are empirical cross-entropies on the same sampled continuations, with each observed token supplied to the model; their near equality does not establish equality of distributions or successful free-running generation. For

the representational test, the affine map from $[x_t/t,\ y_t/t,\ \log t]$ to state-mean activations is fitted on the training measure alone, evaluated at the removed states' coordinates to synthesize centroids the network never produced, and each visiting sequence's activation is assigned to the nearest of the 25 same-time centroids (23 training states plus the two synthesized points); the metric is the fraction assigned to the removed state's own synthesized centroid, at eight residual-stream sites (Figure 5). At first-layer post-attention, all ten two-layer runs identify every visiting context correctly. Of the ten single-layer runs, seven reach 100%, two reach 85.74% and 97.56%, and one reaches 0%. In the two-layer model the first post-MLP site is not exactly at zero (two of ten seeds at 0.55 and 0.23); it is exactly zero in every seed before attention, at LN-final, and after the final MLP.

Forbidden states. The fifteen Walker O models of the main sweep (three architectures, five seeds, $K = 8$, no retraining) are fed counterfactual contexts whose prefix is legal and whose final move steps onto a hole, at $t \in \{3, 5, 7\}$, paired with controls that share the prefix and take a legal final move drawn from the true kernel. An affine map from $[x_t/t,\ y_t/t,\ \log t]$ to first-layer post-attention activations is fitted per run on legal contexts only; its pseudoinverse decodes the counterfactual activations, and the decoded position is snapped to the nearest parity-consistent lattice cell. The next-move distribution emitted at the forbidden state is scored by forward KL against three exact laws: the walker with all holes removed, the walker with only the occupied hole removed, and the true walker at the entry cell one step earlier (stayed put). The same 704 distinct counterfactual paths are evaluated for each model, giving 3,520 model–context evaluations per architecture across five seeds. Within each (architecture, t) cell, evaluations are pooled over seeds ($n = 40$, 320, 3,160); nearest-cell recovery is 100% in every cell, and mean decoding error is lower than on the paired legal control in eight of nine cells. The KL ranking holds in all nine cells: removing only the occupied hole gives 0.0061–0.0187 nats, removing all holes gives 0.0256–0.0529, and ignoring the move gives 0.4179–0.4492. The ignored-move/no-holes KL ratio ranges from 8.05 to 17.19. Even the closest reference remains above the paired legal-control error (2–5 $\times\ 10^{-5}$ nats), so the counterfactual continuation is an approximation.

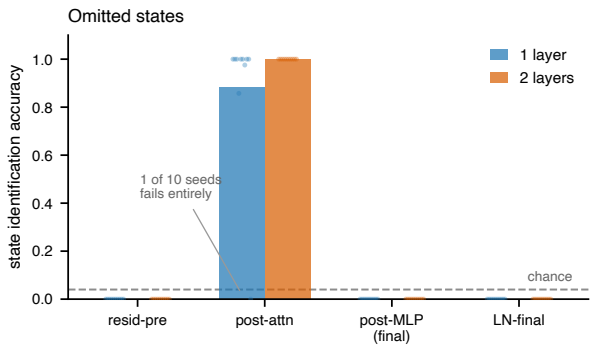

Figure 5: Identification of states omitted from training using an affine coordinate map fitted on observed states. Bars show mean nearest-centroid accuracy and points show individual seeds; the dashed line marks chance ($1/25$). Post-attention denotes the first block, while post-MLP denotes the final block.

D.4 WALKER M: LAYOUT INFERENCE, GENERALIZATION OVER LAYOUTS, AND ITS LOSS DURING TRAINING

Process and training. Each context concatenates eight 12-step return loops on the unbounded lattice ($K = 96$). Layouts contain three distinct masking cells among the 85 positions inside the return horizon $|x| + |y| \leq 6$. We sample a layout uniformly from the training pool for each sequence and draw its loops independently from the exact return-walk distribution, relabeling departures from masking cells as `MASK`. No loop separators or end-of-sequence tokens are added; the first token of a ninth loop supplies the final prediction target. Eight selected locations never host masking cells during training, and we also withhold 10% of the layouts formed from the remaining locations. We train ten seeds on this restricted pool, with $L = H = 4$ and $d_{\text{model}} = 128$ and otherwise the same settings as for Walkers L, D and O, with an extended training budget.

Evaluation. Walker M is evaluated on three layout splits: training layouts, unseen combinations of familiar masking locations, and layouts containing at least one of the eight locations excluded from training (above). Each split uses a fixed sample of $1\,024$ sequences with exact reference laws for every history: the Bayes predictor under the training prior, the Bayes predictor under the full-family prior, which serves as a common reference across splits, and a known-layout oracle that still filters position. On training layouts, we divide expected cross-entropy by conditional entropy under the training-prior Bayes predictor at each position and average over positions, as for L/D/O. Five full-family controls are trained on all $\binom{85}{3} = 98\,770$ layouts. Recovery at excluded cells is the probability the model assigns to `MASK` at departures from such cells, relative to the family reference; it is a probability ratio, not an accuracy. Training trajectories use 25 predeclared checkpoints on the first 256 sequences of each split. All divergences are per-token forward KL from the reference law to the model, in nats.

Behaviour. On training layouts, normalized cross-entropy across the ten restricted seeds averages 1.0016 (1.0013–1.0019), and across the five full-family controls 1.0015 (1.0013–1.0018), where 1 is Bayes optimal (Table 4). The restricted models' excess cross-entropy over the known-layout oracle falls from 0.067–0.068 nats in the first loop to 0.010–0.011 in the eighth, while the family-prior Bayes predictor's curve falls from 0.073 to 0.010 (Figure 6A): the networks extract the layout from observed masks at approximately the rate of exact inference.

Generalization over layouts. Unseen combinations of familiar masking locations show no detectable degradation: for every restricted seed, forward KL to the family reference differs from the training split by at most 1.1×10^{-4} nats per token (Table 4). The common 0.016–0.018 family-reference KL on these two splits largely reflects the difference between the restricted training prior and the full-family prior: against its own training-prior predictor, Walker M's KL is only 0.00051–0.00069. Layouts containing an excluded location are the harder test, because the model has never seen a `MASK` emitted there. Where the family reference expects a `MASK`, the restricted seeds assign

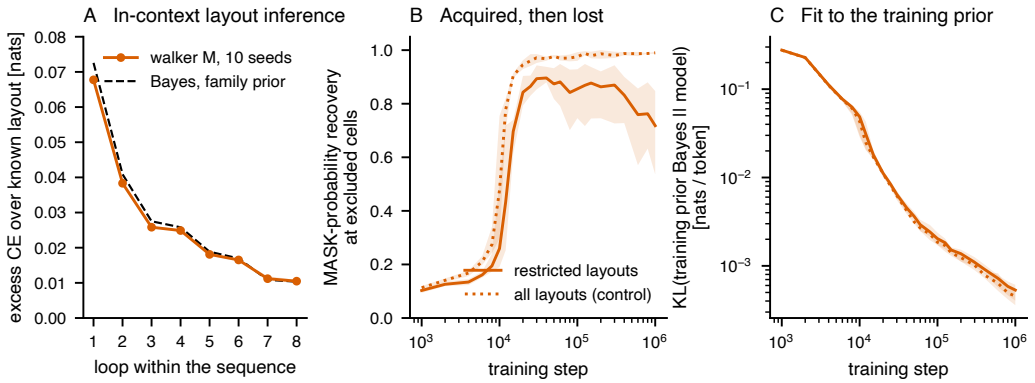

Figure 6: Walker M infers its layout in context at the Bayesian rate and generalizes to excluded locations only transiently. **(A)** Excess cross-entropy over a predictor that knows the layout, per loop of the sequence, averaged over ten restricted seeds on training layouts (solid; shading spans seeds), and for the exact Bayes predictor under the full-family prior (dashed). **(B)** Probability assigned to `MASK` at departures from excluded cells, relative to the family reference, at 25 predeclared checkpoints on a fixed sample of 256 sequences. In (B) and (C), lines show means over ten restricted seeds and five full-family controls; shading spans seeds. The restricted models reach the controls early in training and then lose part of this generalization. **(C)** Over the same checkpoints, the divergence from the Bayes predictor under each model's own training prior keeps falling: the loss in (B) coincides with an ever tighter fit to a prior that never places a masking cell on an excluded location.

Table 4: Walker M against exact Bayesian references. Normalized CE is the expected cross-entropy under each model's training-prior Bayes law, divided by that law's conditional entropy at each position and then averaged; 1 is Bayes optimal. The remaining columns give per-token forward KL in nats to the full-family Bayes predictor. Recovery is the probability assigned to `MASK` at departures from excluded cells, relative to that predictor.

		Training prior	Family reference, KL			
Training pool	Runs	Normalized CE	training	unseen comb.	excluded	Recovery
restricted	10	1.0013–1.0019	0.0156–0.0181	0.0157–0.0181	0.0256–0.0648	0.50–0.85
all layouts	5	1.0013–1.0018	0.0005–0.0006	0.0006–0.0007	0.0012–0.0014	0.99–1.00

50–85% of that probability (mean 72%), compared with 99–100% for the controls, and their divergence from the family reference rises to 0.026–0.065 nats per token, compared with 0.0012–0.0014 for the controls. A location never occupied during training is treated, in part, as one that cannot be occupied.

Acquired, then lost. These final values are the end of a trajectory that we report as a negative result (Figure 6B). On the fixed 256-sequence sample, the restricted seeds reach a mean peak recovery of 92% (89–95%), usually by 150k updates, and then lose part of it: at 10^6 updates they average 72% (54–85%), while the controls average 99%. Nothing in the training objective marks the transition. From 30k to 10^6 updates, divergence from the training-prior predictor falls by a factor of 9.9–17.1 across seeds (Figure 6C), and the unseen-combination split tracks the training split at every checkpoint to within 0.001 nats. The decay is therefore not overfitting in the space of sequences but in the space of layouts: the network fits the layout distribution it was shown ever more exactly, and that distribution never occupies the excluded cells. The early generalization was a transient of an imperfect fit to the prior, and the fully trained model's reading of an excluded cell as one that cannot be occupied is the correct Bayesian conclusion under the training prior. The control shows that the concept itself is not the obstacle: a model that has seen a location host a masking cell keeps it.

Representation. Within a sequence, t resets every twelve tokens. Across time slices $t = 3, \ldots, 9$, we fit shared linear coefficients from $(x_t/t, y_t/t)$ to individual activations, with a separate intercept

for each time. Thus, time can shift the baseline activation but cannot change the spatial map; $\log t$ is not fitted. Observations are weighted so that each (x_t, y_t, t) state contributes equally, and ten-fold cross-validation holds out complete states, including all histories and loops that reach them. At the first post-attention residual, this fit gives pooled held-out $R^2 = 0.720$ on individual activations across the ten restricted seeds (0.597–0.798), compared with 0.959 (0.938–0.976) after averaging activations within each state. The fitted coefficients are the same under these two aggregations; only the score changes because the raw score also counts history-dependent variation within a state. Because a `MASK` hides the move direction, this variation includes both information retained from the path and uncertainty about the realized position used in the fit.

To identify simple sources of this variation, we count the `MASK` tokens observed within the current loop, grouping counts as 0, 1, 2, or ≥ 3. Adding loop- and mask-count-specific offsets to the shared spatial map raises raw held-out R^2 to 0.893 (0.852–0.932); allowing their spatial slopes to vary raises it further to 0.912 (0.882–0.940). The corresponding partial R^2 values are 0.614 and 0.168. Balancing every state–loop–mask-count class equally gives the same conclusion ($0.450 \rightarrow 0.777 \rightarrow 0.799$), so frequent zero-mask histories do not drive it. As a matched complexity control, we permute the joint loop–mask-count labels 199 times within each state, preserving their exact counts, support, and design rank. Across both weightings, all six nested comparisons, and all ten seeds, the smallest real partial R^2 is 0.0081, whereas the largest permuted value is 0.0020. The improvements therefore reflect structure associated with the actual labels rather than model flexibility alone; because those labels are observational and histories within a sequence are dependent, this is not a causal randomization test.

The fitted loop–mask-count offsets mainly translate the spatial map: after centering these class-specific activation shifts, only 0.55% of their squared magnitude lies in the two-dimensional span of the fitted x_t/t and y_t/t coefficient vectors. The smaller slope improvement shows that this separation is approximate, and the remaining within-class variance may encode finer evidence from the path, such as which cells were visited without emitting `MASK`. The map does not depend on the training pool: linear CKA averages 0.988 both among restricted seeds (0.968–0.997) and between restricted seeds and controls (0.966–0.998). The intervention battery of Section 4.2 is run at the same site on 256 sequences, with 300 donor pairs at each of four positions (5, 17, 29 and 41, one in each of the first four loops); donors are taken from another sequence at the same position, and the reference law is the training-prior Bayes predictor for that sequence and position. Pooled over positions, state steering follows the donor in 75.5% of cross-sequence pairs on average across the ten restricted seeds (53.5–86.7%), full activation replacement in 89.7% (82.2–91.8%), replacement by the donor state's mean activation in 91.2% (90.3–92.9%), and norm-matched random steering in 0.3% (0.0–1.4%). State ablation gives 46.1% (36.0–55.4%), compared with 0.3% (0.0–1.2%) for random ablation of the same rank. The five full-family controls give overlapping results, so the causal use of the map does not depend on which layout pool was used for training.

D.5 Additional diagnostics

Context-length robustness. The five walker–architecture pairs are retrained with a context window of 16 (three seeds each) and probed with the same coordinate fit at positions $t \leq 8$ and $t \geq 9$; the $K = 8$ models of the main sweep are passed through the identical pipeline as the baseline. Because exhaustive enumeration is infeasible at length 16, we sample 128 prefixes per state from the exact conditional route distribution and fit the coordinates to individual activations, weighting them so that every state contributes equally. This pipeline differs from the one behind Figure 2A, and its R^2 values are comparable only within this comparison. The ten-fold cross-validation holds out complete states, with separate fits for early and late positions; it does not measure coordinate decoding or transfer a fit from early to late positions. At $t \geq 9$, first-layer post-attention raw R^2 ranges from 0.953 to 0.986 over the five architecture means (0.943–0.987 over individual runs), and exceeds the early-position value in all 15 runs (Figure 7).

Final activations against the next-step target. Figure 8 shows the raw LN-final activations behind the summary statistics of Section 4.1, as the top two principal components plotted against the ground-truth effective field $\boldsymbol{q}_t$ for walkers L and D. The layerwise geometry of the same activations at the post-attention residual, post-MLP residual, and output logits is in Figure 1B; both are single representative runs, and the population statistics over seeds are reported in the main text.

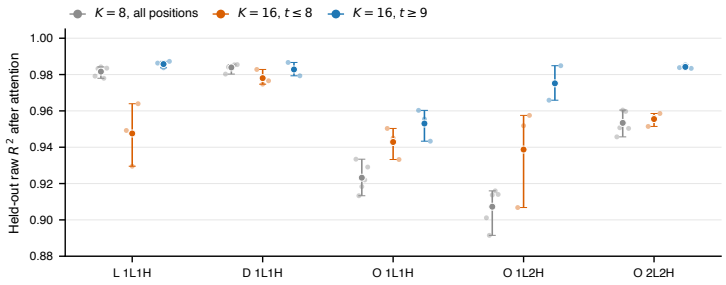

Figure 7: The state coordinates persist when the context window is doubled. Points are seeds; markers and whiskers show means and minima–maxima. All three conditions use the same sampled-route pipeline and fit individual activations with equal total weight per state. The length-16 models use three seeds per architecture; the length-8 baselines use five.

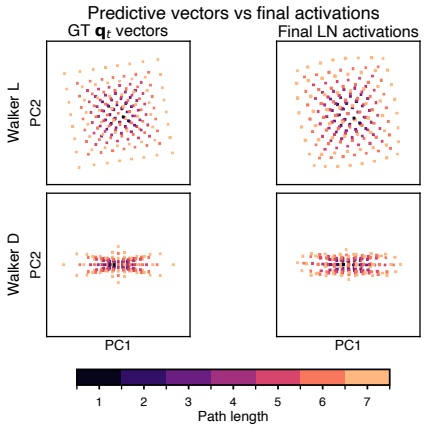

Figure 8: Last-LayerNorm activations (right column) closely resemble the ground-truth next-step targets (left column) for walker L (top) and walker D (bottom). For both walkers the target is the effective field $\boldsymbol{q}_t$ of Equation 8; for walker L $\boldsymbol{q}_t$ is effectively 2D since $q_{uv} = 0$, whereas for walker D it has all three components nonzero. Each point is a distinct path on the lattice, colored by path length. Activations are Procrustes-aligned to the ground-truth target coordinates.

Held-out feature recovery across d_{head}. Figure 9 repeats the encoding analysis of Figure 2A at head widths 32 and 128, again with ten seeds for each of the five walker–architecture pairs. The pattern holds at every width: post-attention R^2 is 0.846–0.986 across all 150 models and exceeds post-MLP R^2 in every one of them, by at least 0.070. After the MLP, R^2 is 0.65–0.82 for Walkers L and D and 0.05–0.83 for Walker O, whose one-layer models vary most across seeds. The earlier single-run sweep in Table 5 covers all $L \in \{1, 2\}$, $H \in \{1, 2\}$ and $d_{\text{head}} \in \{32, 64, 128\}$ for the three walkers; across its 36 models, post-attention R^2 lies in the 0.83–0.99 range.

Feature decodability across d_{head}. Figure 10 reports held-out affine decoding of the two targets from individual residual-stream activations at every hook, for the five walker–architecture pairs of Figure 2 at three head widths and ten seeds each. The state coordinates are decodable after the first attention block in all 150 models (R^2 0.979–1.000). The effective field $\boldsymbol{q}_t$ is not: after the first attention block, held-out R^2 is 0.07–0.77 for Walkers L and D and -0.54–0.30 for Walker O. It becomes decodable after the MLP for Walkers L and D (0.971–1.000), only in the second block for the two-layer Walker O model, and at LN-final in every model (0.84–1.000; all but one run above 0.95). One decrease runs the other way. At LN-final of the one-layer Walker O models, the state coordinates no longer transfer to held-out states: R^2 ranges from -6.18 to 0.94, and 19 of 60 runs fall below zero, although in-sample R^2 is at least 0.945. The two-layer Walker O model retains them (0.941–0.994).

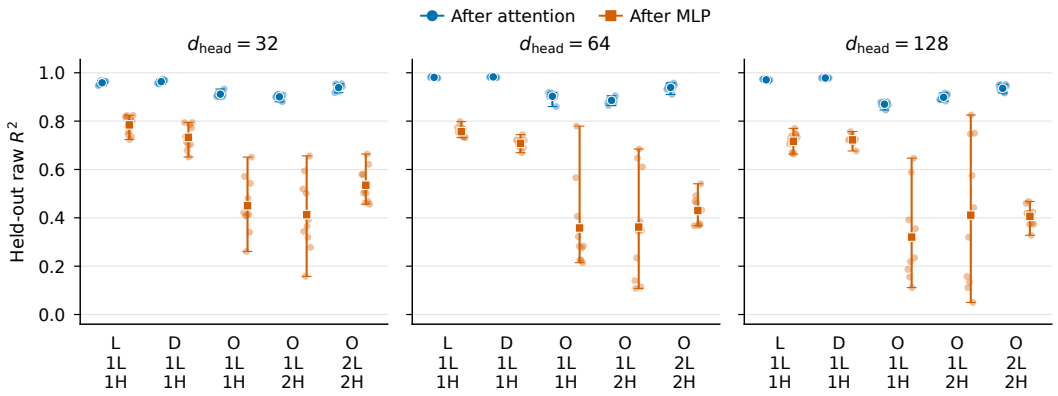

Figure 9: The state coordinates span post-attention activations at every head width, and much less of the post-MLP activations. Held-out encoding R^2 for the affine map from $[x_t/t, y_t/t, \log t]$ to individual first-block activations, with ten held-out-state folds and equal total weight per state, for five walker–architecture pairs at $d_{\text{head}} \in \{32, 64, 128\}$. Small points are seeds; large markers and whiskers show means and minima–maxima over ten seeds. The $d_{\text{head}} = 64$ panel repeats Figure 2A. Table 5 reports single runs of the complete architecture sweep.

Table 5: Individual-activation encoding R^2 from $[x_t/t, y_t/t, \log t]$ across the 36 trained transformer models, using ten held-out-state folds and equal total weight per state. The columns labelled attn and MLP denote the post-attention and post-MLP residual streams of the first block.

			$d_{\text{head}} = 32$		$d_{\text{head}} = 64$		$d_{\text{head}} = 128$	
walker	L	H	attn	MLP	attn	MLP	attn	MLP
L	1	1	0.949	0.847	0.980	0.793	0.967	0.775
L	1	2	0.984	0.733	0.968	0.755	0.937	0.707
L	2	1	0.942	0.854	0.986	0.825	0.952	0.811
L	2	2	0.986	0.791	0.951	0.785	0.933	0.785
D	1	1	0.956	0.781	0.985	0.695	0.981	0.750
D	1	2	0.984	0.726	0.979	0.757	0.972	0.797
D	2	1	0.953	0.802	0.988	0.770	0.973	0.662
D	2	2	0.988	0.699	0.977	0.644	0.962	0.684
O	1	1	0.884	0.397	0.849	0.101	0.837	0.078
O	1	2	0.889	0.368	0.841	0.448	0.900	0.749
O	2	1	0.827	0.718	0.930	0.473	0.925	0.377
O	2	2	0.963	0.534	0.945	0.534	0.946	0.468

Cross-variant CKA across seeds. Table 6 reports all ten model pairs at matched seeds, using first-layer post-attention activations and head width 64. Its ranges summarize variation across seeds; they do not combine measurements from different layers. Matching seeds does not matter: for every pair, the mean CKA over the 90 unmatched seed pairs in Figure 2B differs from the matched-seed mean by at most 0.005.

Pairwise CKA across all individual runs. Figure 11 reports the full pairwise linear CKA matrix between state-mean activations of every trained network in our sweep, without grouping by walker variant. Each row and column is one run, labeled by walker and architecture. The block structure of the matrix reflects the universality result of Section 4.1 at run level: at the post-attention residual stream most pairs are highly aligned, while at the post-MLP residual stream the cross-variant blocks lose alignment.

Layerwise transformation of the fitted feature grids. Figure 12 traces the transformation described in Section 4.1 explicitly, by passing synthetic grids that span the fitted post-attention subspace through the downstream layers. The grids stay close to a two-dimensional sheet throughout, which the LayerNorms and MLP bend into the walker-specific output geometry.

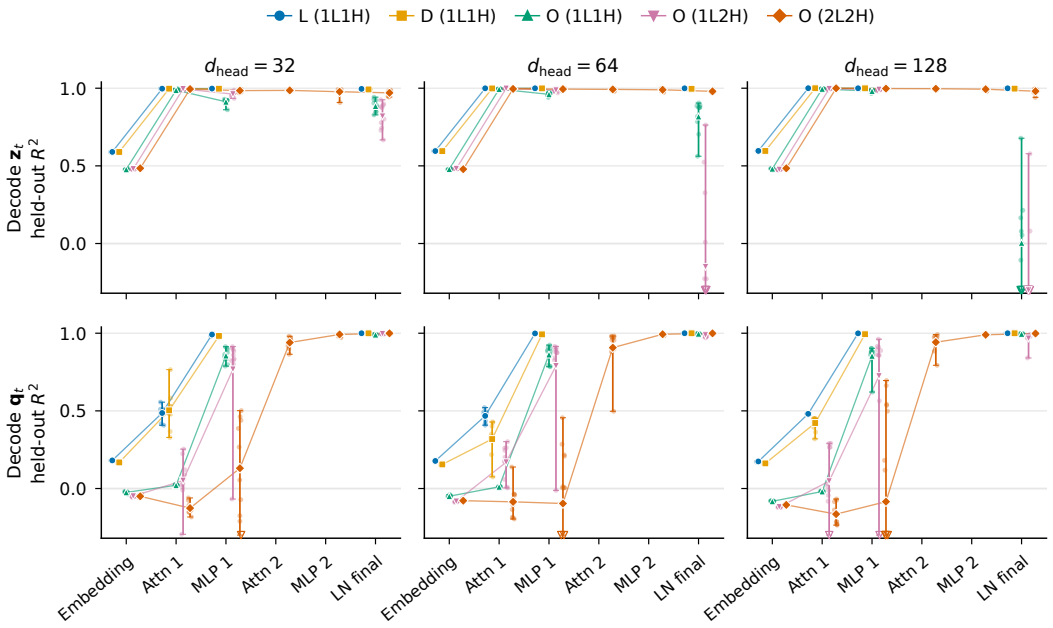

Figure 10: State coordinates are decodable from the first post-attention residual in every model, whereas the effective field becomes decodable only after the MLP or in a later block. Held-out affine decoding R^2 from individual residual-stream activations to the state coordinates $\boldsymbol{z}_t = [x_t/t, y_t/t, \log t]$ (top) or the effective field $\boldsymbol{q}_t$ (bottom), at $d_{\text{head}} \in \{32, 64, 128\}$, for five walker–architecture pairs with ten seeds each. Folds hold out whole states and are stratified by time step; ridge penalties are chosen by nested cross-validation. Small points are seeds; markers and whiskers show means and minima–maxima. Values below -0.3 are drawn at that floor as open triangles. Gaps mark second-layer hooks, absent in one-layer models.

Table 6: Cross-variant linear CKA at the first post-attention residual, head width 64. Each row compares the two models at matched seeds 1–10, using state-mean activations on the pairwise intersection of valid prefixes. Ranges describe variation across seeds, not confidence intervals.

Model A	Model B	Mean lCKA	Seed range
D (1L1H)	O (1L1H)	0.746	0.695–0.788
D (1L1H)	O (1L2H)	0.772	0.734–0.805
D (1L1H)	O (2L2H)	0.970	0.937–0.991
L (1L1H)	D (1L1H)	0.985	0.976–0.997
L (1L1H)	O (1L1H)	0.830	0.771–0.884
L (1L1H)	O (1L2H)	0.852	0.818–0.895
L (1L1H)	O (2L2H)	0.986	0.966–0.994
O (1L1H)	O (1L2H)	0.988	0.938–0.997
O (1L1H)	O (2L2H)	0.845	0.785–0.893
O (1L2H)	O (2L2H)	0.866	0.816–0.931

Top-representation and output-logit PCA dimension. The layerwise pipeline in Figure 12 suggests that the final readout is close to two-dimensional. To compare this low-dimensionality before and after the readout, Figure 13 measures the variance outside the best two-dimensional PCA subspace for the top representation (transformer final LayerNorm activations, or the top recurrent hidden state for RNNs) and for the raw output logits across the transformer and GRU/LSTM sweeps.

D.6 RNN COMPARISON

This appendix gathers the diagnostic plots underlying the cross-architecture comparison summarized in Section 4.6. We train 60 RNNs covering LSTM and GRU cells, $L \in \{1, 2\}$ recurrent layers, hidden

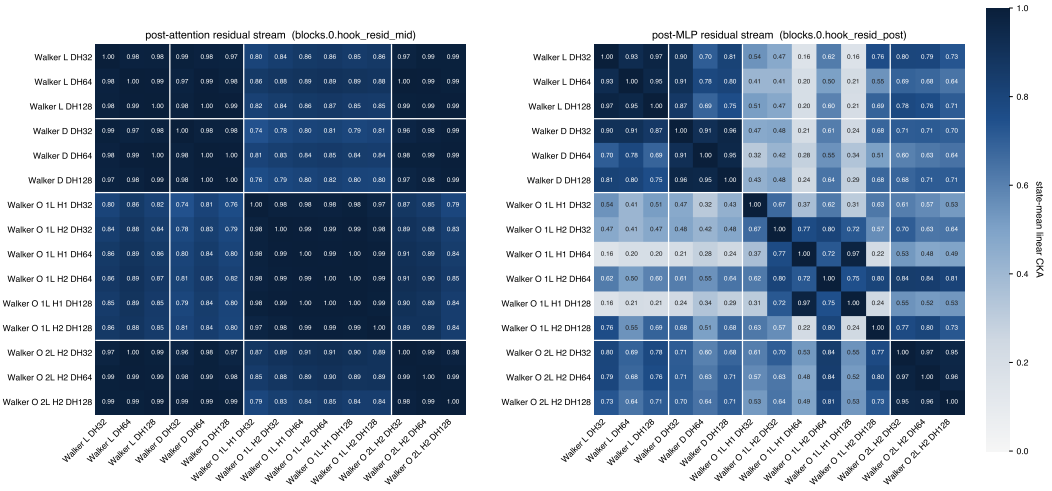

Figure 11: Full pairwise linear CKA matrix between state-mean activations of every individual run in our sweep, at the post-attention residual stream (left) and the post-MLP residual stream (right).

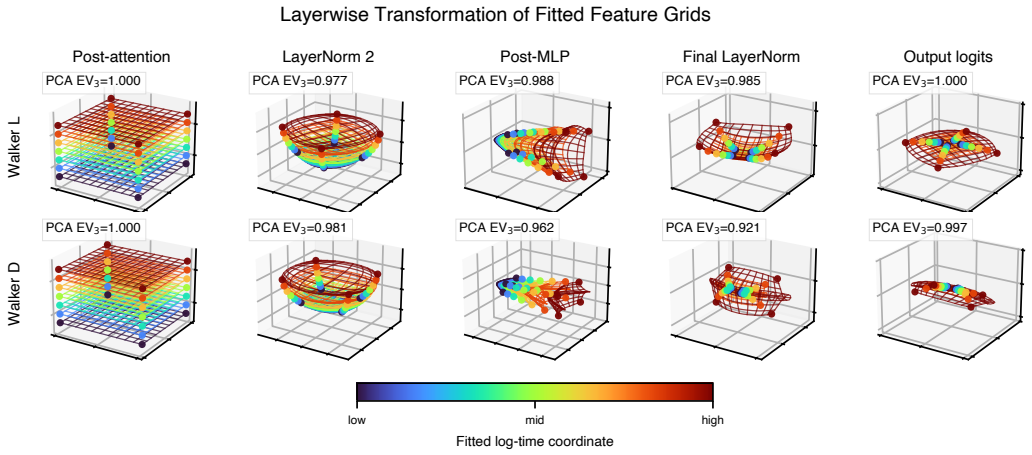

Figure 12: Layerwise transformation of synthetic feature grids. We construct grids spanning the actual datapoints at the post-attention residual stream of Walker L (top row) and Walker D (bottom row), with points colored by the fitted $\log t$ coordinate (low to high, see colorbar). Each subsequent column shows the image of these grids at LayerNorm 2, post-MLP, the final LayerNorm, and the output logits. The annotated PCA EV_3 is the fraction of variance captured by the top three principal components of the activations at each stage.

sizes $H \in \{16, 32, 64, 128, 256\}$, and all three walker variants. All RNNs use the same context length, optimizer, and learning-rate schedule as the transformer sweep, and validation losses are normalized by the same per-position Bayes-optimal floor used for the transformer (Tables 7 and 8).

Predictive performance. All Walker L and D models finish within 2.1×10^{-5} of the normalized Bayes-optimal loss. Walker O is harder at small widths: its excess loss reaches 0.0025 at $H = 16$, but falls below 3.8×10^{-4} for every model with $H \geq 64$ (Figure 14). The corresponding final losses and training trajectories are reported together with the transformer sweep in Section E.3.

Affine transition model. Let $\mathbf{r}_t$ denote a concatenation of everything carried into the next recurrent step: the hidden states of all layers and, for an LSTM, their cell states as well. For each network, we fit the affine transition $\mathbf{r}_t \approx M\mathbf{r}_{t-1} + \mathbf{b}_{c_t}$, constraining the matrix M to be the same for all four move tokens while allowing a separate offset $\mathbf{b}_{c_t}$ for each observed move c_t (here c_t is the observed

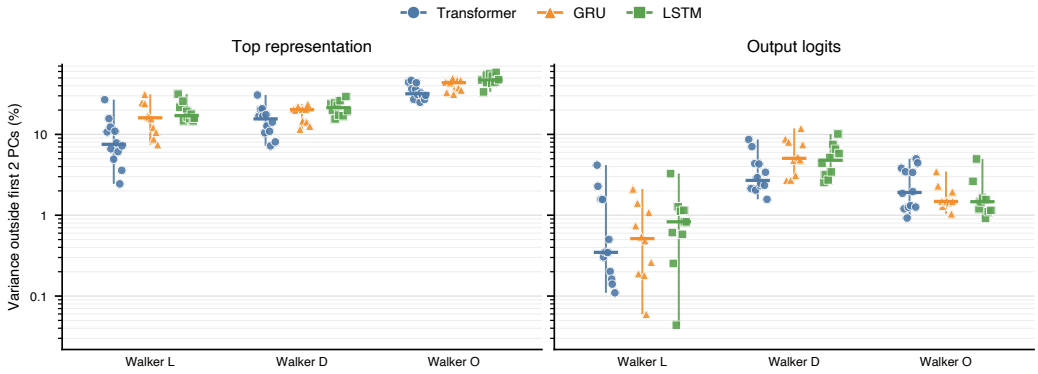

Figure 13: Variance outside the best two-dimensional PCA subspace for the top representation (left: transformer final LayerNorm activations or RNN top hidden states) and raw output logits (right). Each point is one final checkpoint. Color and marker shape denote architecture family; vertical bars show the within-family min–max range for each walker, and horizontal ticks show the median.

move token, not the LSTM cell state). In five-fold cross-validation over transitions, the predicted next recurrent state explains 99.2–99.6% of its variance for Walker L and 96.4–98.8% for Walker D. The shared affine rule fits Walker O transitions less accurately, consistent with obstacles making spatial updates depend more strongly on location. This transition structure is partly inherited from the task: applying the same transition regression to the exact effective field $\boldsymbol{q}_t$, which parameterizes the ground-truth next-move probabilities, gives held-out $R^2 = 0.998$ for Walker L and 0.960 for Walker D. The affine update therefore cannot by itself establish a representation beyond the current prediction. Such additional information is nevertheless present in states with identical next-move probabilities: the recurrent state distinguishes visits to the origin at different times in Walker L, and distinguishes $(0, 0)$ from $(\pm 2, 0)$ in Walker O. These comparisons establish that the information is present; the interventions below establish that the spatial information affects the computation.

Recurrent-state interventions. For each model, the steering vector $\mathbf{b}_\mathrm{R} - \mathbf{b}_\mathrm{L}$ comes from the full-data fit of the shared-matrix transition model above. Under the fitted model, it is the difference between a right and a left move from the same starting state, corresponding to a displacement of two cells to the right. We add it to individual recurrent states after token $t = 2$, 3, or 4, then continue the trained recurrence through token 7 on the unchanged remaining moves. Each intervention time uses up to $4\,000$ sampled valid sequences; at each evaluation position, we retain those whose shifted state $(x + 2, y, t)$ belongs to the enumerated support. We compare the mean forward KL from the exact next-move distributions at the shifted and original positions to the perturbed model's prediction. The counts below refer to these comparisons of means, not to individual-path success rates: 60 comparisons comprise 20 models and three intervention times. Controls include no perturbation and one Gaussian random direction per model, scaled to the shared-fit vector's norm and reused across intervention times.

For Walker L, the shared-fit vector favors the shifted position in all 60 comparisons immediately; for Walker D, it does so in $40/60$. After one update, both walkers favor the shifted position in all 60 comparisons and retain this preference through the remaining context. For Walker O, the corresponding counts are $60/60$ immediately and after one update, $44/60$ after two, and $50/60$ after three; the random controls give $6/60$, $0/60$, $0/60$, and $1/60$, respectively. The preference can disappear at one position and return at the next: for intervention at $t = 3$, the shifted position is favored by $10/20$ models after three updates and by $20/20$ after four.

As an alternative spatial direction, we average recurrent states over all enumerated valid prefixes reaching each (x, y, t), then average their differences over all supported pairs (x, y, t) and $(x+2, y, t)$, giving each pair equal weight. Adding this vector to individual recurrent states also favors the shifted position throughout the continuation for Walkers L and D after one update, and gives the same Walker O counts through three updates. Its immediate effect differs: Walker D favors the original position in all 60 comparisons.

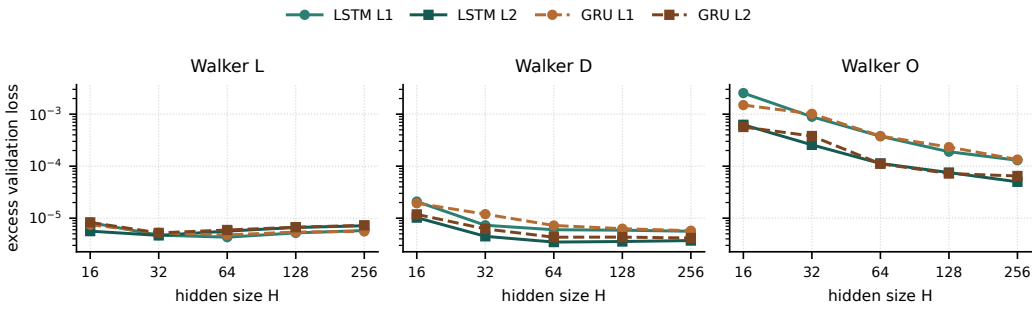

Figure 14: RNN capacity scan: predictive loss. Final validation loss is plotted as excess over the Bayes-optimal value, as a function of hidden size, recurrent cell type, depth, and walker variant. Loss is a model-level quantity, so two-layer models appear once rather than separately for each recurrent layer. All RNN families approach the optimum as width increases; Walker O remains the hardest variant at small widths.

Linear accessibility of the sufficient statistic. We probe each recurrent hidden state in both directions: the *decode* direction maps the activation A to either the 3D state coordinates $\boldsymbol{z}_t = [x_t/t, y_t/t, \log t]$ or the 3D effective field $\boldsymbol{q}_t$, and the *encode* direction is the converse. At the top hidden state, mean held-out decode R^2 across the 20 models per walker is 0.97, 0.96, and 0.94 for $\boldsymbol{z}_t$, and 1.000, 0.999, and 0.985 for $\boldsymbol{q}_t$, on Walkers L, D, and O, respectively. The encode direction is more discriminating: averaged over the 20 models per walker, $\boldsymbol{q}_t$ explains 90%, 42%, and 11% of top-hidden-layer activation variance for Walkers L, D, and O, respectively, while the state coordinates $\boldsymbol{z}_t$ explain 78%, 71%, and 43%. Thus, the statistics are readily recoverable even though they leave substantial hidden-state variance unexplained, most strongly for Walker O—consistent with the diffuse hidden-state geometry shown in Figure 16. The *logits* series in Figure 15 probes the output-layer logits as a sanity check. For Walkers L and D, $\boldsymbol{q}_t$ is a fixed linear transformation of the ground-truth centered log probabilities, so near-optimal model logits should decode it almost exactly, as observed. For Walker O, the target instead regularizes blocked channels with run-specific finite values.

Geometry of the hidden-state manifold. For a representative width $H = 128$, the 3D PCA of the state-mean recurrent hidden representation is a near-2D dome on Walker L, a 3D bowl on Walker D, and a capacity-sensitive shape on Walker O. The pattern reproduces nearly identically between LSTM and GRU, between $L = 1$ and $L = 2$, and within an $L = 2$ stack between layer 0 and layer 1; depth contributes a near-isometric reparametrization rather than a structural change (Figure 16).

Cross-architecture representational similarity. We compute state-mean linear CKA between every transformer residual-stream hook of the canonical $d_{\text{head}} = 64$ run and every per-layer hidden state of every RNN in the sweep, separately for each walker (Figure 17). Three blocks emerge per walker. Within the transformer's own residual stream, lCKA is low—the pre-attention, post-attention, and post-MLP states have substantially different representational geometries, matching the per-stage decomposition that Section 4.1 resolves at the group level. Within the RNN family, state-mean lCKA is uniformly high—every recurrent hidden state across hidden sizes, depths, and the LSTM/GRU split clusters into essentially one form. The off-diagonal RNN-vs-transformer block sits at ≈ 0.5–0.7 for Walkers L and D and lower for Walker O. LayerNorm hooks are excluded since RNNs have no analogue.

E Training Details and Loss Curves

E.1 Hyperparameters

All models are instances of the `HookedTransformer` architecture from the TransformerLens library (Nanda & Bloom, 2022), decoder-only transformers with LayerNorm normalization and ReLU MLP activations. We sweep over architectures with $L \in \{1, 2\}$ layers and $H \in \{1, 2\}$ attention heads, with head dimension $d_{\text{head}} \in \{32, 64, 128\}$; the model dimension is $d_{\text{model}} = H \cdot d_{\text{head}}$ and the MLP width is $d_{\text{mlp}} = 4\, d_{\text{model}}$. The original architecture sweep uses one model per walker–

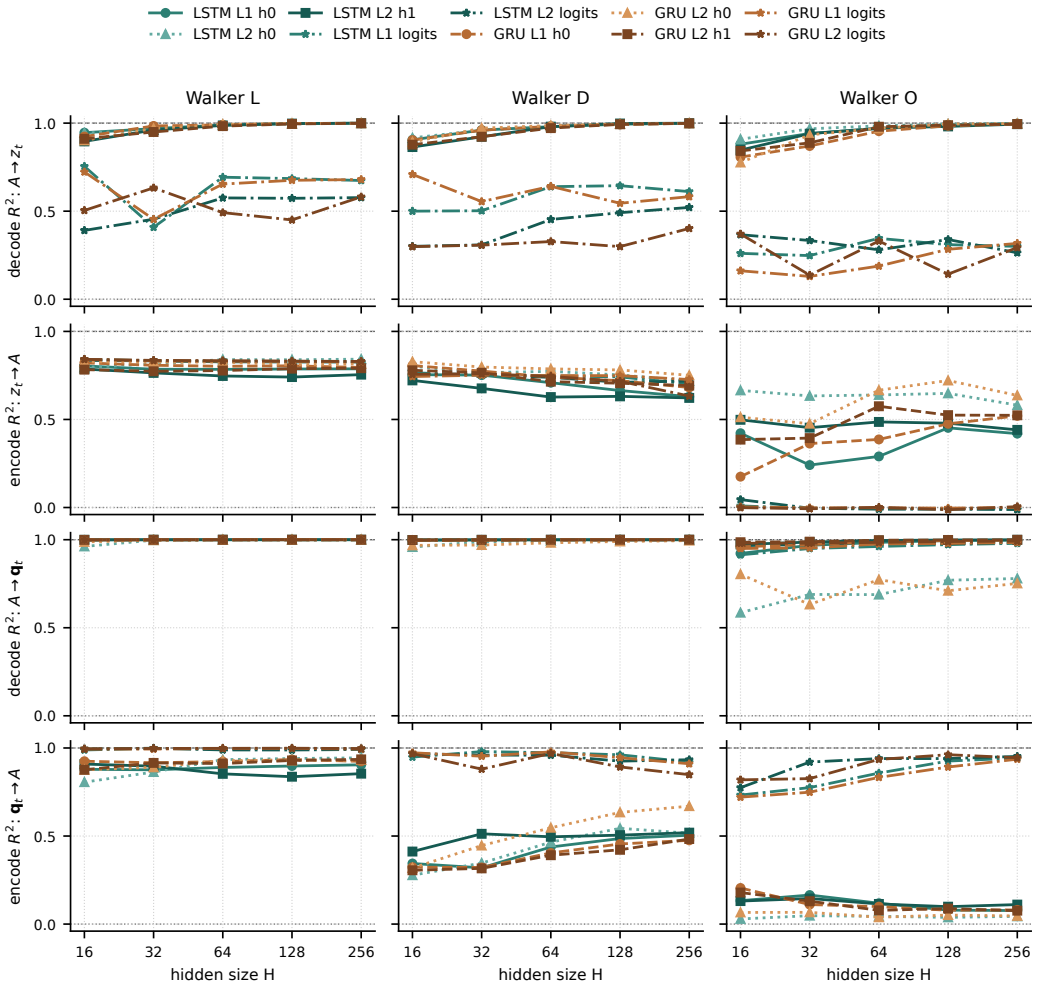

Figure 15: RNN capacity scan: linear accessibility of the state coordinates and the effective field. Each panel shows held-out CV R^2 as a function of hidden size for LSTM and GRU recurrent networks trained on the three walker variants. The target $\boldsymbol{z}_t = [x_t/t, y_t/t, \log t]$ is the vector of state coordinates used in the main text, while $\boldsymbol{q}_t$ is the effective field. Decode rows fit targets from individual activations; encode rows predict individual activations from the targets, with equal total weight per state. For two-layer RNNs, $h0$ and $h1$ denote the first and second recurrent hidden layers; for one-layer RNNs, $h0$ is the only hidden layer. The *logits* series probes the model's output logits as a sanity check: for Walkers L and D, near-optimal outputs decode $\boldsymbol{q}_t$ almost exactly.

architecture pair; the seed and intervention analyses use ten independently trained seeds per pair. The context window is fixed to $K = 8$ tokens. Training ran for 10 000 epochs of 70 minibatches of size 256, that is, 700 000 updates. Walker M models, whose architecture and context length $K = 96$ are given in Appendix D.4, keep this minibatch structure and extend the budget to 14 286 epochs, that is, 1 000 020 updates; the optimizer and learning-rate schedule below are the same. Networks were trained with PyTorch's AdamW optimizer with default `betas` and `eps`—$(0.9, 0.999)$ and 10^{-8}—and zero weight decay. The peak learning rate was 10^{-4}, with a cosine schedule including a 3% warmup that decays to 10% of the peak by the end of training.

E.2 Initialization scale

We retrain the five walker–architecture pairs of Figure 2 with the initialization scale multiplied by $\frac{1}{4}$, $\frac{1}{2}$, 2 or 4 (40 runs in total). We compare the in-sample R^2 of the affine map from the state

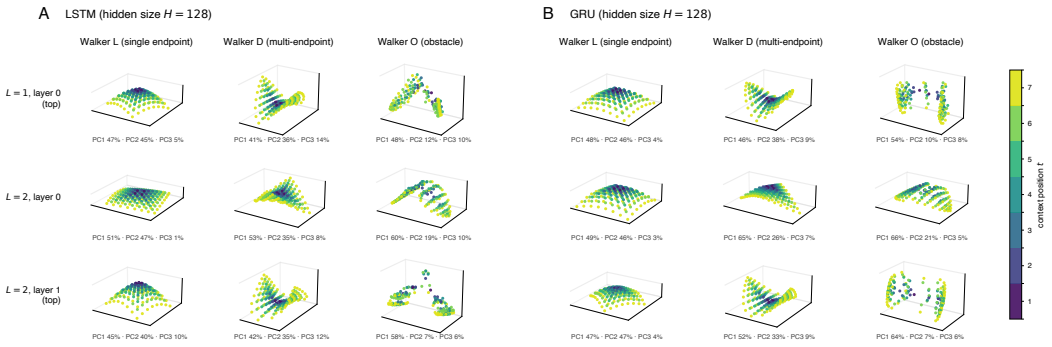

Figure 16: Hidden-state geometry across architectures, depths, and walkers at $H = 128$. **(A)** LSTM. **(B)** GRU. Within each panel, rows are $L = 1$ top hidden ($\equiv$ layer 0), $L = 2$ layer 0, and $L = 2$ layer 1 (top); columns are Walker L, Walker D, and Walker O. Each cell shows the 3D PCA of the state-mean recurrent activation, coloured by context position t, with axes drawn on equal scale in data units so a near-flat manifold renders as flat. The geometry is set primarily by the task: Walker L collapses to a near-2D dome, Walker D forms a 3D bowl, and Walker O is capacity-sensitive. Neither the LSTM/GRU choice nor the depth substantially alters the manifold's intrinsic shape.

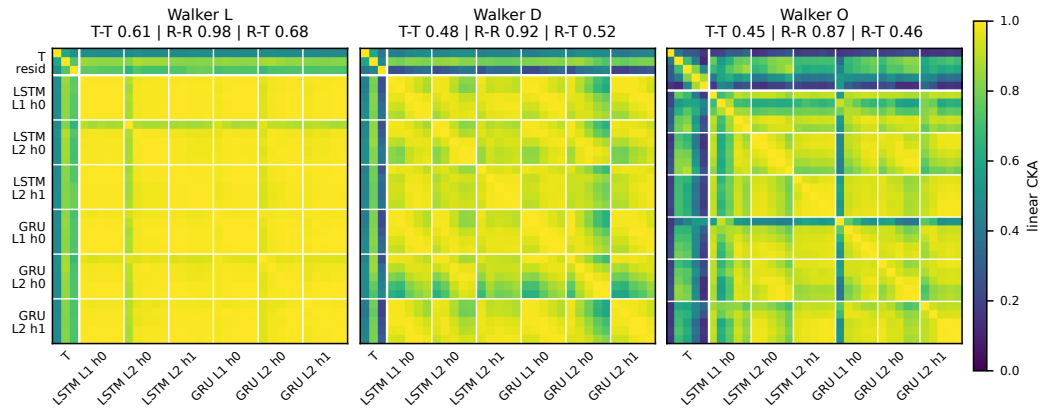

Figure 17: All-vs-all state-mean lCKA, transformer residual streams vs. RNN hidden layers, per walker. Rows and columns of each heatmap are: every transformer residual-stream hook of the canonical $d_{\text{head}} = 64$ run (the first few entries, separated by the white guide lines), followed by every per-layer hidden state of every RNN in the sweep, grouped by (cell type, depth, hidden size, layer index). Three blocks are visible per walker: the transformer's own residual stream is internally dissimilar; the RNN family is internally very similar; the off-diagonal block between the two architectures is moderate (≈ 0.5–0.7 for Walkers L/D, lower for Walker O). LayerNorm hooks are excluded.

coordinates to post-attention activations; this fit is not cross-validated and exceeds the held-out values of Figure 2A by up to 0.014, so we use it only for comparisons across scales. Averaged over the five models, it is 0.942, 0.946 and 0.939 at scales $\frac{1}{4}$, $\frac{1}{2}$ and 2, against 0.948 at the default scale, and falls to 0.755 at scale 4 (range over runs 0.43–0.94). Eight of the ten scale-4 runs end within 0.002 nats of the default-scale loss of the same model, so their weaker geometry (R^2 0.58–0.94; 0.58 and 0.69 for the two O-2L2H runs) is not a failure to fit the task. The other two, both O-1L1H, end 0.020 and 0.044 nats above it and do not converge; one has the lowest R^2, 0.43. Across scales, the R^2 of the affine map from the exact next-step log-probabilities to the LN-final activations falls monotonically, from 0.72 at scale $\frac{1}{4}$ to 0.68, 0.45 and 0.30 at scales $\frac{1}{2}$, 2 and 4.

Table 7: Final validation loss for every trained model, reported as the ratio $\mathrm{CE}_{\mathrm{model}}/H(c_{t+1} \mid s_t)$. At each context position the numerator is the model's cross-entropy averaged over all prefixes, weighted by their probabilities, and the denominator is the exact conditional entropy of the ground-truth process at that position; the table gives the unweighted mean of these per-position ratios, so a value of 1.000 corresponds to a Bayes-optimal predictor. Each column is one of the three walker variants (Section 3.2); each row is one architecture from the appendix sweep ($L \in \{1, 2\}$ layers, $H \in \{1, 2\}$ heads, $d_{\mathrm{head}} \in \{32, 64, 128\}$).

L	H	d_{head}	walker L	walker D	walker O
1	1	32	1.000011	1.000006	1.000138
1	1	64	1.000010	1.000004	1.000044
1	1	128	1.000012	1.000005	1.000033
1	2	32	1.000010	1.000004	1.000042
1	2	64	1.000012	1.000005	1.000035
1	2	128	1.000011	1.000005	1.000038
2	1	32	1.000011	1.000006	1.000036
2	1	64	1.000011	1.000005	1.000023
2	1	128	1.000011	1.000005	1.000021
2	2	32	1.000011	1.000005	1.000023
2	2	64	1.000011	1.000004	1.000024
2	2	128	1.000011	1.000004	1.000019

E.3 Final losses and training curves

Tables 7 and 8 report final validation losses for the transformer and RNN sweeps, normalized by the exact per-position conditional entropy of the ground-truth process. The conditional entropy $H(c_{t+1} \mid s_t)$ is the irreducible uncertainty about the next token given the prefix under the true generative process: it is the cross-entropy attained by a predictor that outputs the ground-truth conditional distribution, and no causal model can do better. We refer to this floor as the *Bayes-optimal* value, so that a dimensionless ratio of 1.000 corresponds to matching the optimal predictor. Figures 18 and 19 show the corresponding validation curves over training. The transformer sweep reaches within 1.5×10^{-4} of the optimum in every case; the RNN sweep follows the same convergence pattern, with the largest residual excess in the smallest Walker O models. At matched seeds, the single-block Walker O models end at a slightly higher loss than the two-block model: for each of O-1L1H and O-1L2H, in all 9 (initialization, data-order) cells of the re-seeded width-64 arm and in all 5 original seeds, by 1.6–7.7×10^{-5} nats; unpaired, the seed-to-seed spread of each architecture ($\approx 3 \times 10^{-4}$ nats) is larger than this gap, so only the paired statement is meaningful.

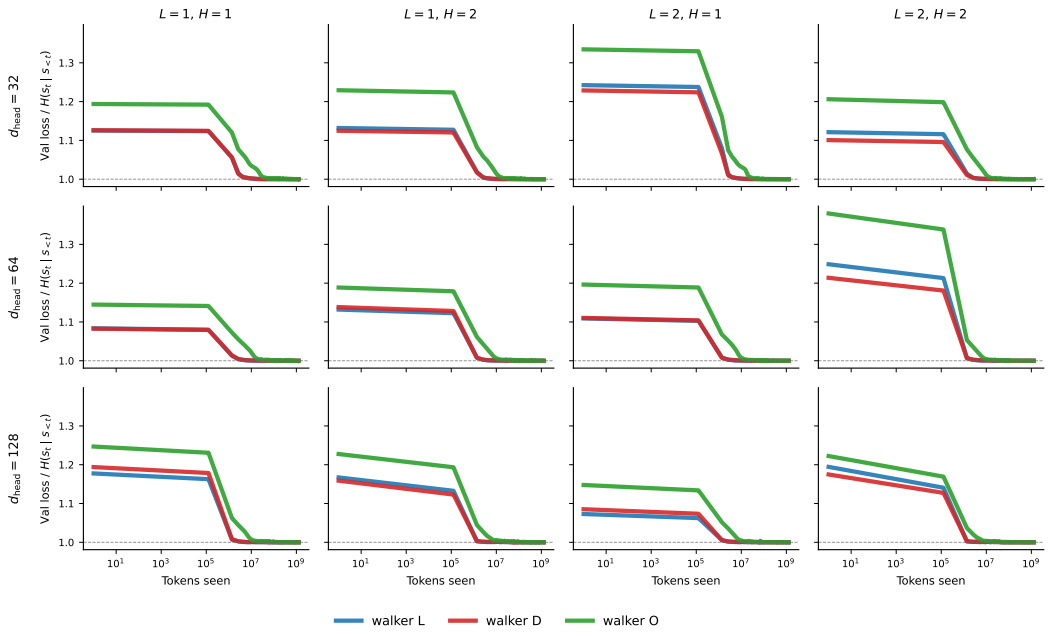

Figure 18: Validation loss vs. tokens seen for the full sweep. Rows are head dimensions ($d_{\text{head}} \in \{32, 64, 128\}$); columns are layer/head combinations. Each panel shows three curves, one per walker variant. The vertical axis is the same dimensionless ratio as Table 7: $\text{CE}_{\text{model}}/H(c_{t+1} \mid s_t)$, averaged over context positions; the dashed line marks the Bayes-optimal value 1.0. The horizontal axis is the cumulative number of training tokens (log scale).

Table 8: Final validation loss for every trained RNN, reported as the same normalized ratio $\text{CE}_{\text{model}}/H(c_{t+1} \mid s_t)$ defined in Table 7. Rows enumerate recurrent cell type, recurrent depth, and hidden size; columns are the three walker variants.

cell	L_{rnn}	H_{hid}	walker L	walker D	walker O
LSTM	1	16	1.000008	1.000021	1.002547
LSTM	1	32	1.000005	1.000007	1.000892
LSTM	1	64	1.000004	1.000006	1.000376
LSTM	1	128	1.000005	1.000006	1.000190
LSTM	1	256	1.000006	1.000006	1.000130
LSTM	2	16	1.000006	1.000010	1.000625
LSTM	2	32	1.000005	1.000004	1.000257
LSTM	2	64	1.000006	1.000003	1.000113
LSTM	2	128	1.000007	1.000004	1.000075
LSTM	2	256	1.000007	1.000004	1.000050
GRU	1	16	1.000007	1.000019	1.001497
GRU	1	32	1.000005	1.000012	1.001017
GRU	1	64	1.000005	1.000007	1.000380
GRU	1	128	1.000005	1.000006	1.000233
GRU	1	256	1.000006	1.000006	1.000134
GRU	2	16	1.000008	1.000012	1.000567
GRU	2	32	1.000005	1.000006	1.000380
GRU	2	64	1.000006	1.000004	1.000111
GRU	2	128	1.000007	1.000004	1.000072
GRU	2	256	1.000007	1.000004	1.000065

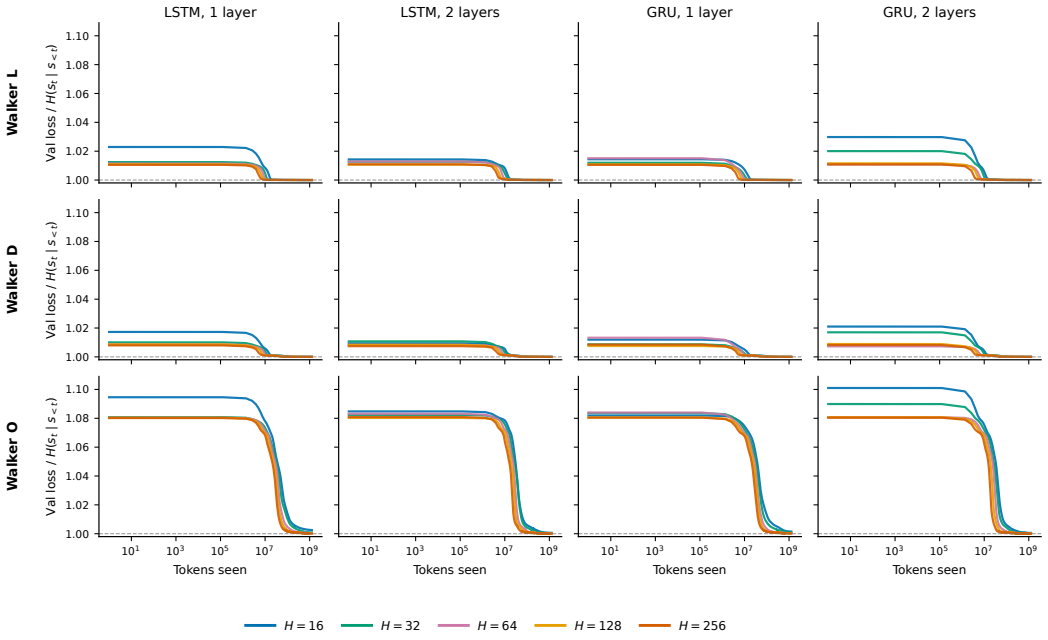

Figure 19: RNN validation loss vs. tokens seen for the full recurrent sweep. Rows are walker variants; columns are recurrent cell and depth. Each panel overlays hidden sizes $H \in \{16, 32, 64, 128, 256\}$. The vertical axis is the same normalized validation-loss ratio used in Tables 7 and 8; the dashed line marks the Bayes-optimal value 1.0.