Delay Embedding Theory of Neural Sequence Models

Mitchell Ostrow [1]   **Adam Eisen** [1]   **Ila Fiete** [1]

Abstract

To generate coherent responses, language models infer unobserved meaning from their input text sequence. One potential explanation for this capability arises from theories of delay embeddings in dynamical systems, which prove that unobserved variables can be recovered from the history of only a handful of observed variables. To test whether language models are effectively constructing delay embeddings, we measure the capacities of sequence models to reconstruct unobserved dynamics. We trained 1-layer transformer decoders and state-space sequence models on next-step prediction from noisy, partially-observed time series data. We found that each sequence layer can learn a viable embedding of the underlying system. However, state-space models have a stronger inductive bias than transformers–in particular, they more effectively reconstruct unobserved information at initialization, leading to more parameter-efficient models and lower error on dynamics tasks. Our work thus forges a novel connection between dynamical systems and deep learning sequence models via delay embedding theory.

1. Introduction

Neural sequence models, specifically transformers and state-space models (SSMs) trained on next-token prediction, have made extraordinary strides in natural language processing (Vaswani et al., 2017; Gu et al., 2021; Orvieto et al., 2023; Gu & Dao, 2023; Lin et al., 2021; Voelker et al., 2019; Gu et al., 2020; 2022). More generally, these models operate over ordered sequences of data, and thus have the potential to be learners of any temporal prediction problem. Yet, transformers have been noted to underperform in continuous time-series prediction (Zeng et al., 2023), an issue that several transformer architecture variants have sought to rectify (Wu et al., 2021; Zhou et al., 2021; Nie et al., 2022). Certain SSMs outperform transformer variants on benchmark time-series prediction tasks (Gu et al., 2021). Furthermore, it is unclear why transformers underperform other models in time-series forecasting.

In this work, we present mechanistic insights into the performance of transformers and SSMs on time-series prediction tasks of well-characterized dynamical systems. We examine their learned representations and quantify their alignment to the dynamical structure of the underlying system. Our results connect neural networks to the theory of delay embeddings in dynamical systems, thereby shedding light on the inductive biases and capabilities of each architecture for time-series prediction.

1.1. Delay Embedding Theory

Delay embedding is a well-known method for reconstructing and characterizing the geometry of chaotic dynamical systems, when the system is only partially observed. Delay embedding a time-series involves stacking time-delayed copies of the observed data into a vector. The most famous theory in the field, Takens' Delay Embedding Theorem, proved that with sufficiently many delays, a delay embedding of a single variable in a multi-variable dynamical system is diffeomorphic to the original dynamics (Takens, 1981). However, this theorem lacks two important components for practical use: how to pick the optimal delay embedding parameters, and an understanding of the role of observational noise (Casdagli et al., 1991). While Takens provided a prescription for a minimal number of delays to reconstruct the attractor, this is not necessarily the optimal delay embedding. Better embeddings integrate enough information to be robust to observational noise, but not too much so that they are overly distorted, thereby hampering downstream prediction. We demonstrate these tradeoffs in Fig.1. Thus, when creating a delay embedding, one must be selective about which components of the history are used. Methods to pick appropriate delays include autocorrelation analysis, mutual information, persistent homology, and more (Packard et al., 1980; Gibson et al., 1992; Kennel et al., 1992; Fraser & Swinney, 1986; Tan et al., 2023).

Transformers and SSMs can both be viewed as delay embeddings, as they operate over a time history and are capable of inferring latent variables (Piantadosi, 2023). However, the behavior of these sequence models depends on how they combine information across time. Transformer-based language models often sparsely combine inputs representing past states via a learned attention mechanism (Li et al., 2023) while structured state space sequence models (SSMs)
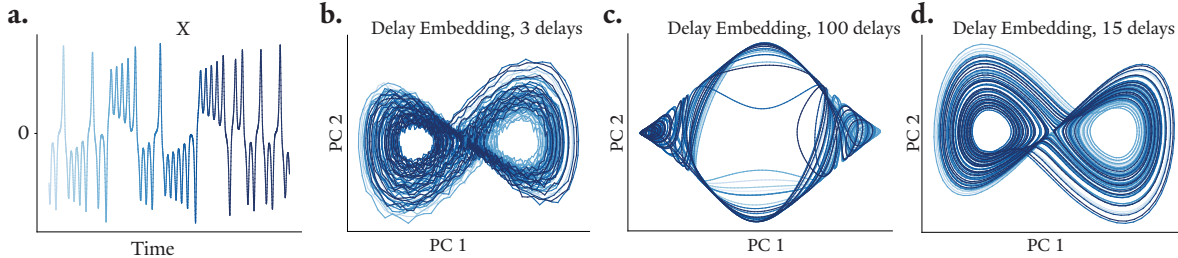
Figure 1. Delay Embeddings. **a.** Noisy data from the x dimension of the Lorenz attractor, on which our models are trained. **b.** Visualization of the top two Principal Components of a delay embedding with too few delays. Here, noise is amplified and the attractor is distorted. **c.** As in **b**, with too many delays. Here the attractor is folded into too many dimensions, making the data harder to model. **d.** In the intermediate, the embedding both reduces noise and has a geometry that reflects the original space (visualized in Appendix A.1).

are designed to memorize as much of the past inputs as possible (Voelker et al., 2019; Gu et al., 2020; 2022). Here, we apply the delay embedding perspective on transformers and SSMs to better understand the inductive bias of each architecture. In particular, we study the performance and dynamics of one-layer transformers and SSMs (the Linear Recurrent Unit (Orvieto et al., 2023)) on a noisy, partially observed, chaotic dynamics prediction task. By forging the connection between delay embedding theory with sequence prediction in deep learning, we hope to establish a relationship that will mutually benefit research in both deep learning and dynamical systems theory.

1.2. Contributions

We characterize the embedding properties of 1-layer SSMs and transformers, showing that SSMs have a stronger inductive bias for delay embeddings, which leads to better attractor reconstructions and lower error on a chaotic prediction task. However, we also show that SSMs contain a large amount of redundancy, which excessively deforms the attractor and makes the model more sensitive to observational noise. While transformers do not have this inductive bias, we find that they are able to successfully learn a viable delay embedding with sufficient training.

2. Methods

We study one-layer sequence models, which consist of the following layers, in order: an encoding matrix, layer normalization, the sequence layer, layer normalization, and a three-layer MLP. We study one form of each class for simplicity: a GPT-style decoder-only transformer, and the linear-recurrent unit (LRU, (Orvieto et al., 2023)). For the fairest comparison between architectures, we study transformers with positional embeddings applied *only* within the softmax function of self-attention. Positional embeddings provide temporal information, breaking the permutation invariance of the transformer inputs to enable higher performance. Not including positional embeddings outside the

softmax function, on the other hand, improves the quality of the manifold reconstructions. Embeddings were learnable.

For a given sequence of inputs U, positional embeddings P, present time point T, our self-attention layer is written as

$$o_T = \text{SoftMax}[(U_{t \leq T} + P_{t \leq T})^T W^{qk}(u_t + p_t)]W^{ov}U_{t \leq T} \tag{1}$$

Notably, taking positional embeddings out of the output (i.e., $W^{ov}(U_{t \leq T} + P_{t \leq T})$ does not negatively affect the performance, as the dynamics of the Lorenz attractor is time-invariant (it is an autonomous system). For convenience, we write the key and query matrices together as W^{qk} and the output and value matrices together as W^{ov}.

The Linear Recurrent Unit layer (Orvieto et al., 2023) is a discrete linear dynamical system:

$$x_{t+1} = Ax_t + Bu_t \tag{2}$$

$$o_t = C\text{Re}(x_t) + Du_t \tag{3}$$

In particular, A is a diagonal matrix with complex eigenvalues initialized uniformly on a disk within the unit circle of the complex plane. This gives the model rotational dynamic properties, thereby allowing each input to be moved to a different subspace and be preserved. Because the input and output are real, the complex component of x is discarded before a linear map to the output.

Our models were trained on next-step prediction for a single observed variable of the 3-dimensional, chaotic Lorenz attractor (see Appendix A.1). We simulated 2000 trajectories of the system for 600 timesteps using the simulation timestep dt $= 0.01$ (Fig. 1a). Each trajectory had a unique initial condition. We removed the first 100 timesteps to eliminate transients. We added in i.i.d. Gaussian noise of zero mean and variance 0.1 to the data at each timestep. We trained our model with Adam for 1000 epochs. For our analysis, we only studied models that reached a Mean Absolute Standardized Error (MASE) < 1. The MASE is
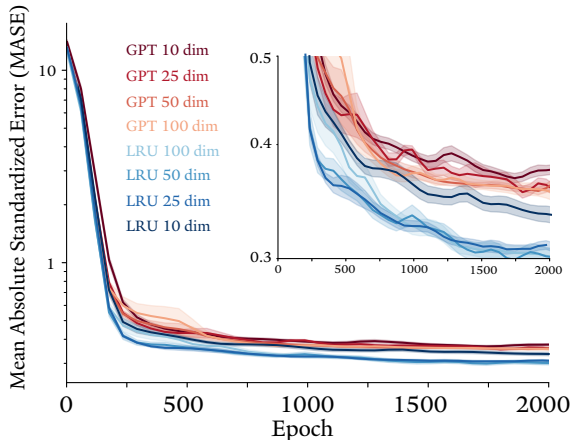
Figure 2. Learning curves of each architecture, inset zoomed in. Faded area around each curve indicates standard error.
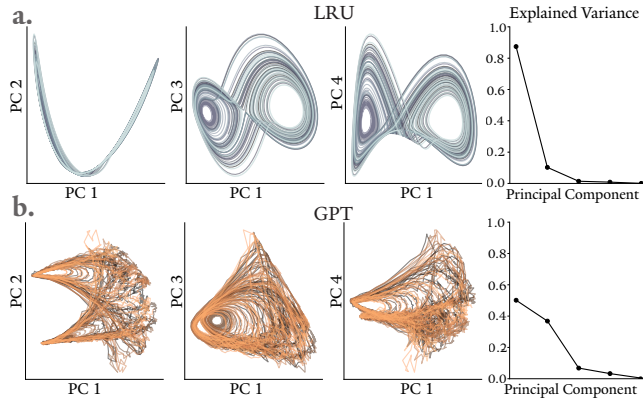

Figure 3. Sample top 4 Principal Components after training, of each sequence layer output, colored by trajectory.

the Absolute Error, normalized by the Persistence Baseline:

$$\text{MASE}(x_t, \hat{x}_t) = \frac{|x_t - \hat{x}_t|}{|x_t - x_{t-1}|} \qquad (4)$$

$\text{MASE} \geq 1$ indicates that the model has captured no predictive information. At the time of submission, we trained approximately 25 networks of each architecture and dimensionality (10, 25, 50, and 100). We collected a similarly sized dataset for different values of observed noise (0.05 and 0.0), and displays these results in Appendix 8.

2.1. Measuring Delay Embedding Quality

We quantified how well the sequence layer outputs operated as delay embeddings via three methods (futher detailed in Appendix 7):

Decoding hidden variables　We trained linear and nonlinear (MLP) decoders to predict the two unobserved dimensions of the 3-dimensional attractor, and measured the test R^2.

Measuring Smoothness　Because a diffeomorphism is a smooth transformation, neighborhoods in one space should map onto neighborhoods in the other space. We measured this by identifying the fraction of overlap between the twenty nearest neighbors in the embedding and the true space.

Measuring Unfolding　Lastly, we measure how well the embedding lends itself to prediction, via the conditional variance of the future data given the embedding, $\sigma_\tau^2(x) = \text{Var}(x(t + \tau)|o_t)$ where $o(t)$ is the model hidden state. We provide implementation details in the appendix. We calculate the average of σ_τ^2 over all data points, and average over τ from 1 to 10 steps in the future.

3. Results

We began our analysis by inspecting the learning curves of each model, plotting the performance across training in Fig. 2. LRU models across all dimensionalities outperformed all GPT models in terms of MASE, albeit by a small margin (final performance in Appendix Fig. 8).

To visually inspect the quality of embeddings, we plotted the top 4 Principal Components of two sample models that solve the noiseless task with MASE 0.06 in Fig. 3 (LRU in **a**, GPT in **b**). Two observations are immediately evident: (1) The LRU embedding is more visually appealing, and (2) the butterfly shape emerged only in the LRU–this suggests that the LRU generated a more faithful delay embedding. However, in the first panel, PCs 1 and 2 suggest that this attractor is quite deformed. On the other hand, the transformer embedding is much less appealing. However, the two lobes of the attractor are identifiable in the first 2 PCs, and the dimensionality of the embedding (Participation Ratio of 2.47) is closer to the true attractor dimension (approximately 1.93) whereas the SSM embedding's dimension is 1.29. This larger dimensionality may lead to increased noise robustness of the transformers (see Appendix, Fig. 9).

We applied each of the embedding metrics to the systems every 50 epochs during training, and plotted the results in Fig. 4, averaging over runs and separated by dimensionality and architecture. Across all metrics, we found that the LRU consistently started off with superior embedding quality. This indicates that the architecture has a powerful inductive bias for delay embedding. Importantly, each metric improved across training, demonstrating that the embedding can be optimized for better performance. The transformers, on the other hand, started off with much worse embedding metric performance, but gradually approached the embedding quality seen in the LRU models. We correlated each met-
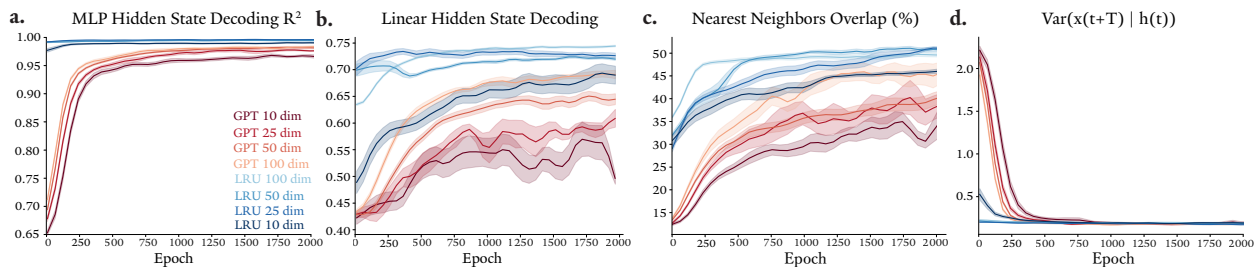
Figure 4. Delay embedding metrics across training, colored by architecture and dimension. **a**. MLP decoding of unobserved variables, test R^2. **b**. Linear decoding test R^2. **c**. Neighbors overlap fraction between full dynamic state and embedding. **d**. Conditional variance of future data given the embedding, averaged over future time steps from 1 to 10. Faded area around each curve indicates standard error.

ric to prediction performance, finding a robust correlation between prediction quality and each of nonlinear decoding (correlation of 0.76), linear decoding (correlation of 0.56), and nearest neighbor overlap (correlation of 0.64, see Appendix A.4).

4. Conclusion

In this study, we demonstrated that the inductive bias of SSMs leads to improved embedding reconstruction, which was correlated with better performance on time series prediction. We found that SSMs have slightly higher performance on a dynamical systems prediction task, but were more sensitive to noise. Furthermore, for similar embedding dimensionality, the use of positional embeddings increases the parameter count of transformers relative to SSMs. This suggests that SSMs may be preferred in the low-data, low-compute regime. While our models and task were simplified, our study identifies a generic property of how time series are combined in each architecture, which is relevant for any application of these models to time series prediction. In future work, we plan to study how transformers and selective SSMs (Gu & Dao, 2023) select their delays.

References

Casdagli, M., Eubank, S., Farmer, J. D., and Gibson, J. State space reconstruction in the presence of noise. *Physica D*, 51(1):52–98, August 1991.

Fraser, A. M. and Swinney, H. L. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A Gen. Phys.*, 33(2):1134–1140, February 1986.

Gibson, J. F., Farmer, J. D., Casdagli, M., and Eubank, S. An analytic approach to practical state space reconstruction. *Physica D: Nonlinear Phenomena*, 57(1-2):1–30, 1992.

Gu, A. and Dao, T. Mamba: Linear-Time sequence modeling with selective state spaces. December 2023.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polyno-mial projections. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1474–1487. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/102f0bb6efb3a6128a3c750dd16729be-Paper.pdf.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. *Int Conf Learn Represent*, abs/2111.00396, October 2021.

Gu, A., Goel, K., Gupta, A., and Ré, C. On the parameterization and initialization of diagonal state space models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 35971–35983. Curran Associates, Inc., 2022.

Kennel, M. B., Brown, R., and Abarbanel, H. D. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, 45(6): 3403–3411, March 1992.

Li, Z., You, C., Bhojanapalli, S., Li, D., Rawat, A. S., Reddi, S. J., Ye, K., Chern, F., Yu, F., Guo, R., and Kumar, S. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. In *The Eleventh International Conference on Learning Representations*, February 2023.

Lin, T., Wang, Y., Liu, X., and Qiu, X. A survey of transformers. *arXiv e-prints*, pp. arXiv:2106.04554, June 2021.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, September 2022.

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. Resurrecting recurrent neural

networks for long sequences. *ICML*, abs/2303.06349, March 2023.

Packard, N. H., Crutchfield, J. P., Farmer, J. D., and Shaw, R. S. Geometry from a time series. *Physical review letters*, 45(9):712, 1980.

Piantadosi, S. Modern language models refute chomsky's approach to language. *Lingbuzz Preprint, lingbuzz*, 7180, 2023.

Sugihara, G., May, R., Ye, H., Hsieh, C.-h., Deyle, E., Fogarty, M., and Munch, S. Detecting causality in complex ecosystems. *science*, 338(6106):496–500, 2012.

Takens, F. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, pp. 366–381. Springer Berlin Heidelberg, 1981.

Tan, E., Algar, S., Corrêa, D., Small, M., Stemler, T., and Walker, D. Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(3):032101, 03 2023. ISSN 1054-1500. doi: 10.1063/5.0137223. URL https://doi.org/10.1063/5.0137223.

Uzal, L. C., Grinblat, G. L., and Verdes, P. F. Optimal reconstruction of dynamical systems: a noise amplification approach. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 84(1 Pt 2):016223, July 2011.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Voelker, A., Kajić, I., and Eliasmith, C. Legendre memory units: Continuous-time representation in recurrent neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/952285b9b7e7a1be5aa7849f32ffff05-Paper.pdf.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. *Adv. Neural Inf. Process. Syst.*, pp. 22419–22430, June 2021.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, volume 37 of *AAAI'23/IAAI'23/EAAI'23*, pp. 11121–11128. AAAI Press, February 2023.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence Time-Series forecasting. *AAAI*, 35(12):11106–11115, May 2021.

A. Supplementary Information

A.1. Lorenz attractor equations

$$\frac{dx}{dt} = \sigma(y - x) \tag{5}$$

$$\frac{dy}{dt} = x(\rho - z) - y \tag{6}$$

$$\frac{dz}{dt} = xy - \beta z \tag{7}$$

Where $\sigma = 10$, $\rho = 28$, $\beta = 8/3$. The fractal dimension of the attractor, measured using the Kaplan-Yorke dimension, is approximately 2.06. The dimension of the attractor, which we computed via participation ratio, is 1.986.
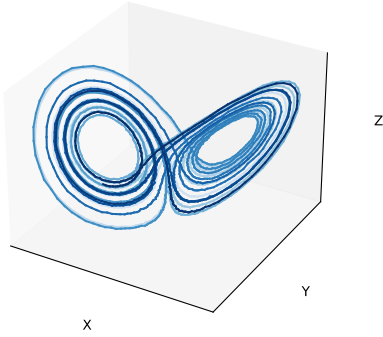

Figure 5. 3-dimensional visualization of the Lorenz attractor. Simulated with noise, colored by time.

A.2. Embeddings before training

In Fig. 6, we visualize the embeddings of each sequence layer before training, when driven with a noiseless input. While neither looks much like the lorenz attractor, and are both quite low dimensional, it is evident that the LRU has a more similar appearance, with the two lobes evident in the 2nd plot.

A.3. Metrics

A.3.1. PARTICIPATION RATIO

The Participation Ratio is a continuous measure of dimensionality, derived from Principal Components Analysis. While it is not equivalent to the fractal dimension of an attractor, which is typically measured via the Lyapunov exponents and is much more challenging to compute from data, it is useful to quantify dimensionality without setting
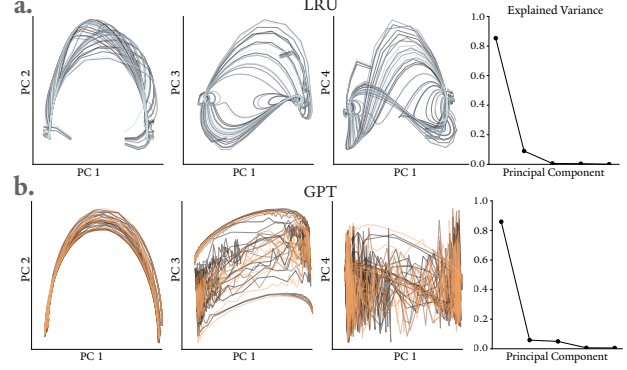

Figure 6. Untrained trajectories from each architecture, the same models as in Fig. 3.

arbitrary thresholds on the explained variance. Given the eigenvalues λ_i of the correlation matrix, XX^T, the participation ratio is calculated as

$$p = \frac{(\sum_i \lambda_i)^2}{\sum_i (\lambda_i)^2} \tag{8}$$

A.3.2. CONVERGENT CROSS MAPPING

Convergent Cross Mapping (CCM, (Sugihara et al., 2012)) was developed as a measure of causality between dynamical systems, but here we chose to use it to measure continuity of the embedding, as it uses similar underlying principles. Given simultaneously recorded time series data from two dynamical systems (x and y), CCM constructs a delay embedding of each, then uses the k-nearest neighbors (with k equal to the embedding dimensionality) in one system to predict the value of the state in the other system. More specifically, given a particular time point $x(t)$, the k-nearest neighbors of the first system, $U(x)$, are identified. These points are then mapped to the embedding $y(t)$ yielding an equivalent set of neighbors, $U(y)$, (given the one-to-one mapping), and the prediction is made via their average:

$$\hat{y}(t) = \frac{1}{k} \sum_{i \in U(y)}^{k} y_i \tag{9}$$

A.3.3. NEIGHBORS OVERLAP

We also implemented a stronger metric of continuity via the k-nearest neighbors, which we call the Neighbors Overlap. For each given point, mapped one-to-one via the embedding: $y = f(x)$, we compute the nearest neighbors of the true data space and the embedding separately: $U(x)$, $V(y)$. Then, we identify the time indices of each neigbor: T_u, T_v. The metric averages the fraction of overlap between these index

sets across each point in the dataset:

$$\text{Overlap}(X, Y) = \frac{1}{n} \sum_i^n \frac{|T_u(i) \cap T_v(i)|}{k} \qquad (10)$$

Where $|T_u(i)| = |T_v(i)| = k = 20$, $|X| = |Y| = n$.

A.3.4. UNFOLDING METRIC

We implemented the embedding complexity metric from (Uzal et al., 2011). For noisy data, this metric characterizes the noise robustness of the embedding, and more generally calculates the complexity of the model required for next-step prediction. Here, we explain the motivation for the metric and detail the computational steps required to implement it.

In Casdagli et al. 1991 (Casdagli et al., 1991), the authors suggest that predictive value is a good quantity to optimize for in an embedding. The authors define the predictive value of a reconstructed coordinate y as the conditional probability density on the time-series values T time-steps ahead:

$$p(x(t + T)|y(t))$$

where $x \in \mathbb{R}^D$ is the observed time-series and $y \in \mathbb{R}^d$ is the reconstructed coordinate. As noted by the authors, this quantity is independent of the predictive estimation procedure, as it captures what can be predicted about $x(t + T)$ from $y(t)$ with a perfect estimation procedure. The authors then go on to suggest that the variance of this distribution, given by

$$\text{Var}(x|y) = \int x^2 p(x|y)dx - \left(\int x p(x|y)dx \right)^2$$

is a reasonable quantity to optimize for. Given that the ideal predictor is given by $\hat{x} = E(x|y)$, the above variance is the mean-square prediction error of the ideal predictor, and thus presents a lower bound on the mean-square prediction error of any predictor.

Building on this idea, Uzal et. al. (Uzal et al., 2011) define the unfolding metric, which aims to estimate this variance. Given a time series dataset, the unfolding metric calculates two values for each time step based on its k-nearest neighbors. The first measures the variance of these points in the input space as time progresses. The latter is the volume of these points in the embedding space.

The conditional variance of future time steps, given the embedding, is approximated using the nearest neighbors:

$$E_k^2(T, \tilde{x}) \equiv \frac{1}{k + 1} \sum_{\tilde{x}' \in U_k(\tilde{x})} [x'(T) - u_k(T, \tilde{x})]^2 \qquad (11)$$

where $\tilde{x}$ is the delay embedded initial condition, $x'(T)$ is the future value of the true time series x corresponding to initial

delay embedded condition $\tilde{x}$, $B_\epsilon(\tilde{x})$ is a Gaussian ball with standard deviation ϵ around $\tilde{x}$, $U_k(\tilde{x})$ is the neighborhood of $k + 1$ points containing $\tilde{x}$ and its k neighbors (and is an approximation of $B_\epsilon(\tilde{x})$, and the mean of the embedding is computed as:

$$u_k(T, \tilde{x}) = \frac{1}{k + 1} \sum_{\tilde{x}' \in U_k(\tilde{x})} x'(T) \qquad (12)$$

Then, the overall conditional variance is computed by averaging E_k^2 over the first p timesteps after the initial condition:

$$\sigma_k^2(\tilde{x}) = \frac{1}{p} \sum_{j=1}^p E_k^2(T_j, \tilde{x}) \qquad (13)$$

where the T_j's index the timesteps.

This is done for each individual data point, and we report the average across the attractor. We can also normalize this by the average volume of the whole attractor, which is equivalent to weighting the metric by the ergodic measure of the attractor. As suggested by Uzal et al. (2011), we used $k = 3$ and $p = 10$.

A.4. Relation of each metric to performance

Here, we demonstrate that in the noisy case, multiple embedding metrics are related to the model's predictive capacity. In Fig. 7, we scatter each metric for the noisy data, with noise variance $\sigma^2 = 0.1$. We observe a robust correlation for the nonlinear decoding accuracy, linear decoding accuracy, and neighbors overlap. This strengthens our hypothesis that a stronger embedding improves prediction performance, and provides insight into the superior predictive capabilities of the LRU models.

As can be seen in Fig. 7f, we do not observe any correlation between performance and embedding complexity. This is likely because we utilize sufficiently wide MLPs for prediction, implicitly limiting the necessary complexity of the hidden embeddings. In future work, we will restrict the model expressivity and seek to identify a connection between embedding complexity and prediction performance.

A.5. Change in performance due to noise

To assess the robustness of each model architecture to observational noise applied i.i.d to each time point, we trained models with noise variances of $\sigma^2 \in \{0.0, 0.05, 0.1\}$. We display the performance of each model in Fig. 8. We also train baseline models which we call 'Delay MLPs'–these are MLPs that operate over explicit delay embeddings. We simulated these with MLP widths ranging from 10 to 100, with a delay interval of 1 and a number of delays of sizes $\{10, 25, 50, 100\}$. We find that the LRU models perform comparably to the Delay MLPs, and significantly better
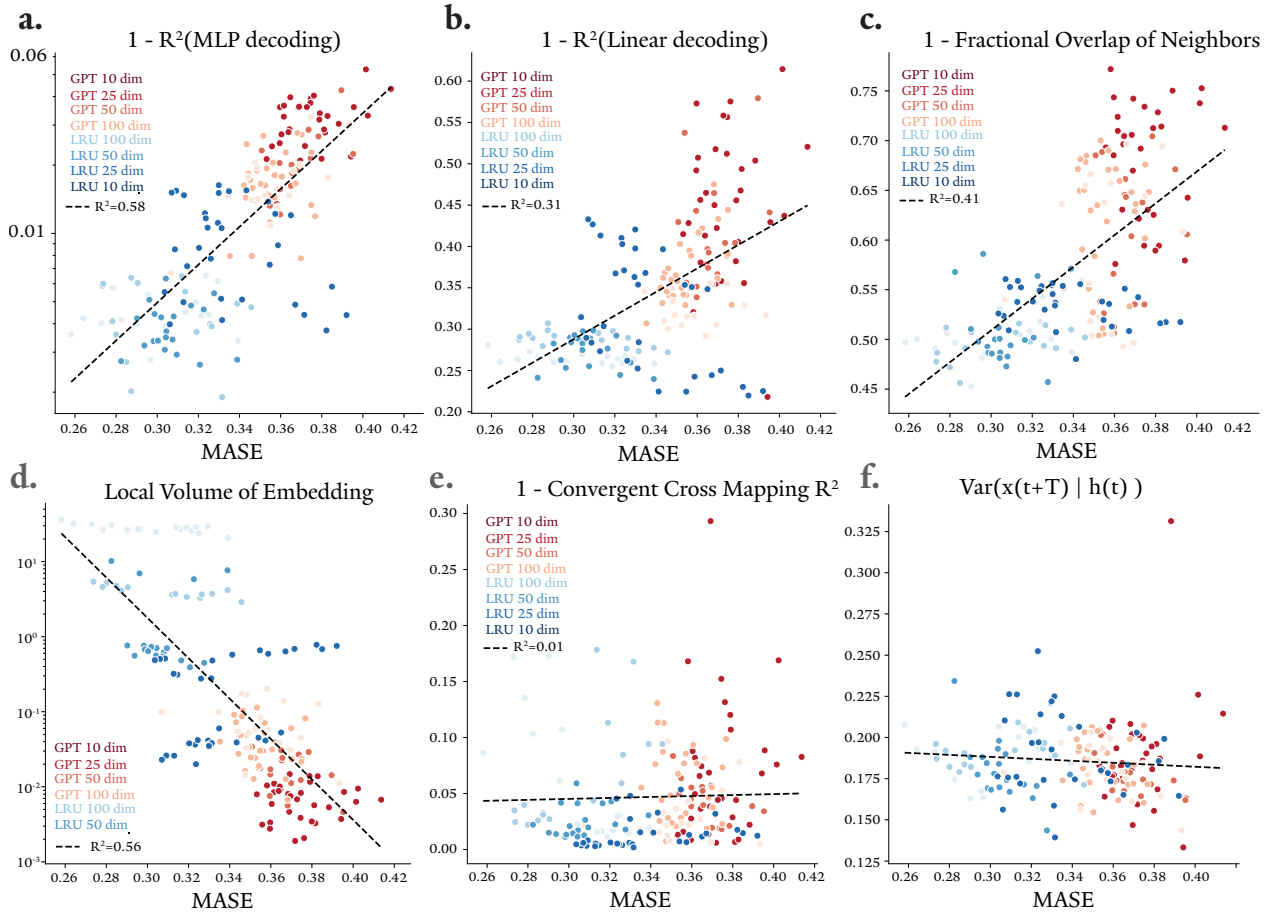
Figure 7. Relation of each embedding metric with performance. Each dot is an individual trained neural network, colored by architecture and dimensionality. x-axis on all plots is the validation Mean Absolute Standardized Error (MASE) after training. Dotted line indicates regression line of best fit, with R^2 coefficient listed in each legend **a.** MASE against nonlinear decoding accuracy R^2, plotted as $1-$ accuracy, due to logarithmic improvement. **b.** Likewise, MASE against linear decoding accuracy, demonstrating that better unfolding also improves performance. **c.** Relationship of MASE with fractional overlap of the 20 nearest neighbors in each embedding space (full data space versus output of the sequence layer). **d.** Relationship of the MASE with the local volume of the embedding, measured as the average distance of the 20 nearest neighbors in the embedding space from the present point. **e.** Relationship with the Convergent Cross Mapping Score, detailed in Appendix 7. **f.** Relationship with conditional variance, a measure of the unfolding complexity of the attractor, detailed in Appendix .
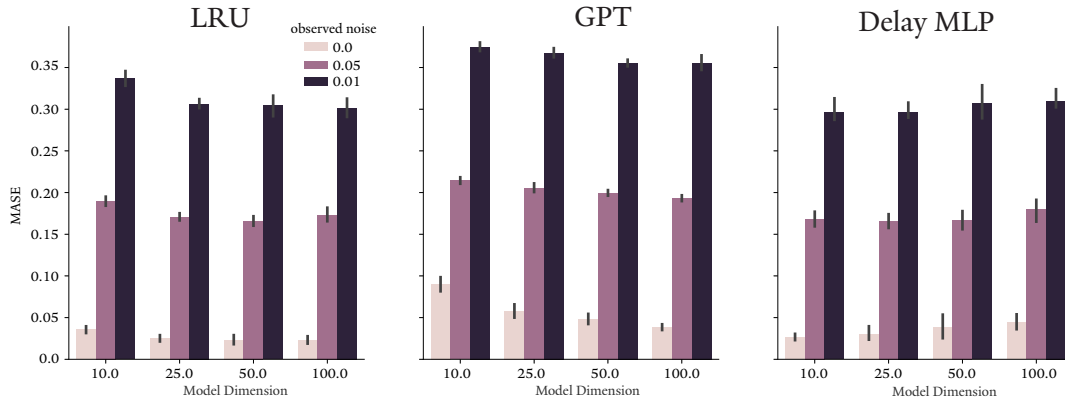
Figure 8. Performance of each model across different dimensionalities and variance of observational noise. On the right, the baseline MLP over the delay embedding is displayed. Bars indicate standard error

than GPTs across all noise levels. However, in Fig. 9, we measure the percent change in MASE when the noise is increased from 0.0 to 0.1, which shows that the LRU is more susceptible to noise than the transformer. This corroborates with results from Fig. 3 which show that the LRU model is more highly folded and lower dimensional than the transformer.
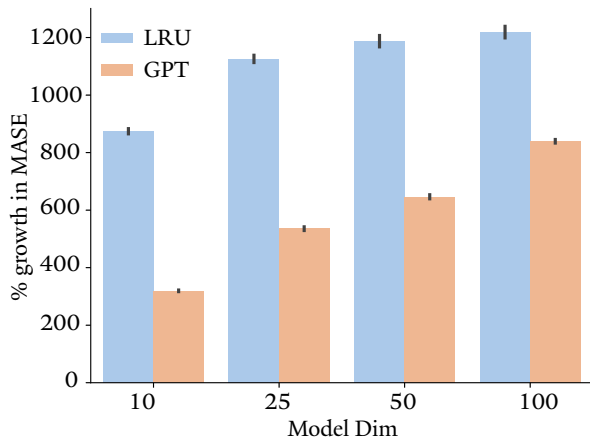

Figure 9. The percent growth in the MASE for the LRU and GPT when noise is increased to 0.1 from 0.0. Bars indicate standard error.