Beyond Geometry: Comparing the Temporal Structure of Computation in Neural Circuits with Dynamical Similarity Analysis

Mitchell Ostrow, Adam Eisen, Leo Kozachkov, Ila Fiete
Department of Brain and Cognitive Sciences, MIT
Cambridge, MA, USA
{ostrow,eisenaj,leokoz8,fiete}@mit.edu

Abstract

How can we tell whether two neural networks utilize the same internal processes for a particular computation? This question is pertinent for multiple subfields of neuroscience and machine learning, including neuroAI, mechanistic interpretability, and brain-machine interfaces. Standard approaches for comparing neural networks focus on the spatial geometry of latent states. Yet in recurrent networks, computations are implemented at the level of dynamics, and two networks performing the same computation with equivalent dynamics need not exhibit the same geometry. To bridge this gap, we introduce a novel similarity metric that compares two systems at the level of their dynamics, called Dynamical Similarity Analysis (DSA). Our method incorporates two components: Using recent advances in data-driven dynamical systems theory, we learn a high-dimensional linear system that accurately captures core features of the original nonlinear dynamics. Next, we compare different systems passed through this embedding using a novel extension of Procrustes Analysis that accounts for how vector fields change under orthogonal transformation. In four case studies, we demonstrate that our method disentangles conjugate and non-conjugate recurrent neural networks (RNNs), while geometric methods fall short. We additionally show that our method can distinguish learning rules in an unsupervised manner. Our method opens the door to comparative analyses of the essential temporal structure of computation in neural circuits.

1 Introduction

Comparing neural responses between different systems or contexts plays many crucial roles in neuroscience. These include comparing a model to experimental data (as a measure of model quality, Schrimpf et al. [2020]), determining invariant states of a neural circuit (Chaudhuri et al. [2019]), identifying whether two individual brains are performing a computation in the same way, aligning recordings across days for a brain-machine interface (Degenhart et al. [2020]), and comparing two models (e.g. to probe the similarity of two solutions to a problem, Pagan et al. [2022]). Current similarity methods include R^2 via Linear Regression, Representational Similarity Analysis, Singular Vector Canonical Correlation Analysis, Centered Kernel Alignment, and Procrustes Analysis (Schrimpf et al. [2020], Kriegeskorte et al. [2008], Raghu et al. [2017], Williams et al. [2021], Duong et al. [2022]). Crucially, these all compare the *geometry* of states in the latent space of a neural network.

Identifying shared neural computations between systems calls for methods that compare similarity at the level of the fundamental computation. In the brain, computations are instantiated by the emergent dynamical properties of neural circuits, such as fixed points, invariant and fixed point manifolds,

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

https://doi.org/10.52202/075280-1467

limit cycles, and transitions between them (Amari and Arbib [1977], Hopfield [1984], Hopfield and Tank [1986], Seung [1996], Zhang [1996], Hahnloser et al. [2000], Burak and Fiete [2009], Wang [2002], Sussillo and Abbott [2009], Churchland et al. [2012], Chaudhuri et al. [2019], Vyas et al. [2020], Khona and Fiete [2022].Geometric similarity metrics fall short in two manners when they are applied to dynamical systems. First, response geometries can differ due to sampling differences (Chaudhuri et al. [2019]) or other nonfundamental causes even though the dynamics are topologically equivalent. In such cases, measures of state-space geometry do not fully capture the core similarities between two systems, a false negative situation. Conversely, systems may exhibit distinct dynamics over similar geometries in the state space (Galgali et al. [2023]). In this case, geometric measures may fail to distinguish two distinct systems, a false positive situation. Therefore, we suggest that dynamical similarity should be the preferred lens through which to compare neural systems.

In dynamical systems, a key notion of similarity is called topological conjugacy: the existence of a homeomorphism that maps flows of one system onto flows of the other. When two systems are conjugate, they have the same qualitative behavior. Given two dynamical systems $f : X \to X$ and $g : Y \to Y$ with invertible mapping $\phi : X \to Y$, conjugacy is defined as:

$$g \circ \phi = \phi \circ f \tag{1}$$

Here, we develop a data-driven method called Dynamical Similarity Analysis (DSA), which combines work from two previously-unrelated fields, machine learning for dynamical systems and statistical shape analysis. In brief, DSA returns a similarity metric describing how two systems compare at the level of their dynamics. First, DSA uses the insight that appropriate high-dimensional embeddings can be used to describe a highly nonlinear system in a way that is globally linear. The features that enable this transformation are referred to as Koopman Modes (Mezić [2005]) and can be identified via the Dynamic Mode Decomposition (DMD) (Snyder and Song [2021], Brunton et al. [2021]). Subsequently, DSA employs a statistical shape analysis metric to compare vector fields within the Koopman Mode embeddings, thereby assessing the similarity between the systems' dynamics. This shape analysis measures how far a pair of systems are from being homeomorphic. Unlike most spatial shape analyses, which require averaging over multiple noisy trials in the same condition to arrive at one vector for each condition-time point (but see Duong et al. [2022]), small amounts of noise create a rich representation of neural dynamics in the estimated dynamics matrix. The importance of flows in response to intrinsic (natural) or induced perturbations in capturing dynamics has been highlighted in Seung [1996], Yoon et al. [2013], Chaudhuri et al. [2019], Galgali et al. [2023], as their evolution over time can uniquely identify dynamic structure.

Our results demonstrate that DSA effectively identifies when two neural networks have equivalent dynamics, whereas standard geometric methods fall short. DSA therefore has great relevance for neuroscience, providing another method to validate the fit of a model to the neurophysiology data. It additionally opens the door to novel data-driven analyses of time-varying neural computation. We expect our method to be interesting to computational neuroscientists due to its theoretical richness, valuable to deep learning researchers as a theoretically backed and rigorously justified tool through which to interpret and compare neural networks, and useful for experimental neuroscientists due to its simplicity of implementation and ease of interpretation.

Contributions We develop a general method, DSA, for comparing two dynamical systems at the level of their dynamics. To do so, we introduce a modified form of Procrustes Analysis that accounts for how vector fields change under orthogonal transformations. We apply our method in four test cases, each demonstrating novel capabilities in neural data analysis: **(a)** Our method can identify dynamical similarities underlying systems with different geometries, which shape metrics cannot. **(b)** Conversely, DSA can distinguish systems with different dynamics despite similar geometries, which shape metrics cannot. **(c)** We show that our method empirically identifies topological similarities and differences between dynamical systems, as demonstrated by invariance under geometric deformation and variance under topological transformation. **(d)** We demonstrate how to use DSA to disentangle different learning rules in neural networks without supervision, by comparing how representations change across training. [1]

[1] https://github.com/mitchellostrow/DSA/

https://doi.org/10.52202/075280-1467

2 Methods

2.1 Theoretical Background: Koopman Operators and Dynamic Mode Decomposition (DMD)

A common approach to modeling nonlinear dynamics is to approximate them with linear systems, as those are more tractable and interpretable. One standard method involves finding fixed points of the dynamics, and then locally linearizing the system around each fixed point, to determine whether infinitesimal changes diverge from or converge to the fixed point. Another is to describe nonlinear dynamics via locally linear models, a separate one for each point in state space. These methods do not generate a description of the global dynamics. The former method also requires injecting perturbations, which may not be possible for experimental systems and pre-collected datasets.

The field of dynamical systems theory suggests a different approach, on which a new class of data-driven methods are based. Specifically, Koopman operators theoretically embed nonlinear systems into infinite-dimensional Hilbert space, which permits an exact and globally linear description of the dynamics (Koopman [1931]). Practical applications use the Dynamic Mode Decomposition (DMD) (Schmid [2010]) to create a finite-dimensional embedding that approximates the Koopman operator (Brunton et al. [2016a, 2017], Snyder and Song [2021], Dubois et al. [2020]). The DMD identifies a linear transition operator for the dynamics: $X(t + \Delta t) = AX(t)$, where X comprises a data matrix (consisting of functions of the observed states of the dynamical system). When applied to a sufficiently rich X, the DMD can capture global dynamical structure. A key challenge of the method is in constructing an appropriate space for X which is closed under the operation of A (a Koopman invariant subspace, Brunton et al. [2016b]). We employ the Hankel Alternative View of Koopman (HAVOK, Brunton et al. [2017], Arbabi and Mezić [2017]), which constructs X as a function of the delay-embedding of the observed variables (a Hankel Matrix), such that one column of $X = [-x(t) - -x(t-\tau) - ... - x(t-p\tau)-]^T$, where p is the number of included delays and τ is the time-step between each measurement. The delay-embedding approach eliminates the difficulty of learning a nonlinear transformation of the data, and improves estimation capabilities over partially-observed systems (Kamb et al. [2020]). The latter capability arises from Takens' delay embedding theorem, which states that a delay embedding is diffeomorphic to the original system–that is, a partially-observed system can be fully reconstructed using a sufficient number of delays (Takens [1981]). However, our method could be used with a range of embedding algorithms in place of delay embeddings.

2.2 Dynamical Similarity Analysis (DSA)

Now, we introduce *Dynamical Similarity Analysis*, a general method that leverages advances in both Statistical Shape Analysis and Dynamical Systems Theory to compare two neural systems at the level of their dynamics. Simply put, we use Koopman operator theory to generate a globally linear embedding of a nonlinear system, via delay embeddings as in HAVOK, then use DMD to define the linear transition operators in this space. Finally, we apply a novel extension of the Procrustes Analysis shape metric to the resultant DMDs from two systems. This set of steps constitutes DSA.

Suppose we have two dynamical systems defined by the following equations:

$$\dot{\mathbf{x}} = f(\mathbf{x}, t) \quad \mathbf{x} \in \mathbb{R}^n \qquad\qquad \dot{\mathbf{y}} = g(\mathbf{y}, t) \quad \mathbf{y} \in \mathbb{R}^m \tag{2}$$

We sample data from each system $X \in \mathbb{R}^{c \times k_1 \times t_1 \times n}$ and $Y \in \mathbb{R}^{c \times k_2 \times t_2 \times m}$. Here, c indicates the number of conditions observed, which in neuroscience and deep learning refer to different input stimuli or contexts. k_i indicates the number of trials per condition, and t_i indicates the number of time steps observed per trial. Note that the observations must be sampled uniformly in time (Takens [1981]). Next, we produce delay-embedded Hankel tensors with a lag of p: $H_x \in \mathbb{R}^{c \times k_1 \times (t-p-1) \times np}$ and likewise for H_y. Using the Hankel Alternative View of Koopman (HAVOK) approach (Brunton et al. [2017]), we flatten all but the last dimension of H and fit reduced-rank regression models with rank r to the eigen-time-delay coordinates of the data, where the target is the coordinates of the next time step ($V'_{x,r}$):

$$V'_{x,r} = A_x V_{x,r} \text{ where } H_x^T = U_x \Sigma_x V_x^T \text{ and } V_{x,r} = V_x[:, 1:r] \tag{3}$$

Only the rank of the HAVOK models explicitly need to be the same so that the two DMD matrices can be compared. Note that the eigen-time-delay coordinate formulation is equivalent to PCA whitening (Supplementary Information Section 12), which Williams et al. [2021] highlights to be important. The predictive capability of our model (via MSE, R^2, AIC, or others) can be used to identify the optimal rank r and number of delays p. In our experiments, hyperparameters for the DMD were chosen to ensure both sufficient capability to fit the dynamics in question as well as tractable computation times. Fitting the HAVOK model to each system returns our DMD matrices: $\mathbf{A}_x, \mathbf{A}_y \in \mathbb{R}^{r\times r}$, which we compare using the modified Procrustes Analysis algorithm detailed next.

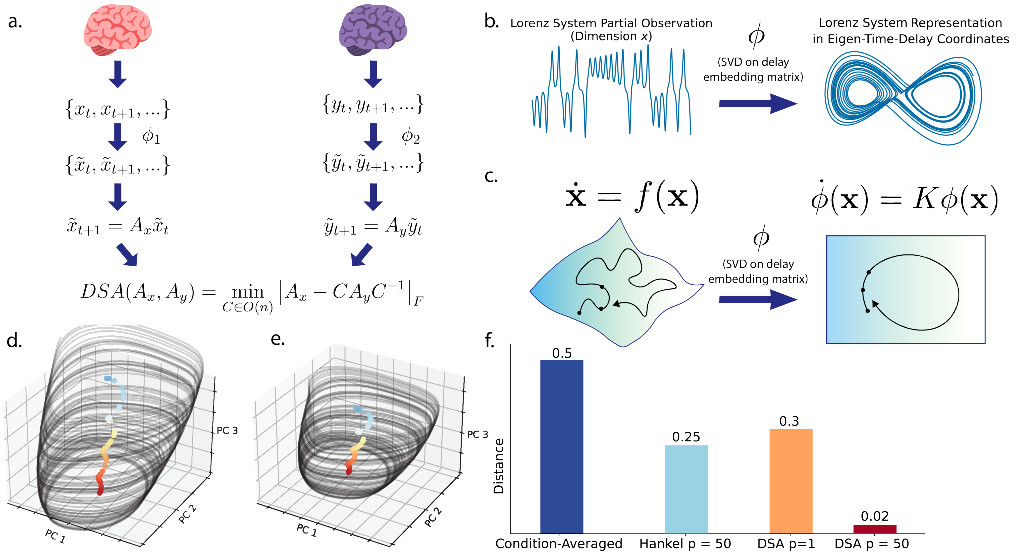
a.
$\{x_t, x_{t+1}, ...\}$
ϕ_1
$\{\tilde{x}_t, \tilde{x}_{t+1}, ...\}$
$\tilde{x}_{t+1} = A_x\tilde{x}_t$
$\{y_t, y_{t+1}, ...\}$
ϕ_2
$\{\tilde{y}_t, \tilde{y}_{t+1}, ...\}$
$\tilde{y}_{t+1} = A_y\tilde{y}_t$
$DSA(A_x, A_y) = \min_{C\in O(n)} |A_x - CA_yC^{-1}|_F$
b.
Lorenz System Partial Observation (Dimension x)
ϕ
(SVD on delay embedding matrix)
Lorenz System Representation in Eigen-Time-Delay Coordinates
c.
$\dot{\mathbf{x}} = f(\mathbf{x})$
$\dot{\phi}(\mathbf{x}) = K\phi(\mathbf{x})$
ϕ
(SVD on delay embedding matrix)
d.
PC 1
PC 2
PC 3
e.
PC 1
PC 2
PC 3
f.
Distance
0.5
0.25
0.3
0.02
Condition-Averaged
Hankel p = 50
DSA p=1
DSA p = 50

Figure 1: **Schematic and example of DSA**. **a.** Schematic of the DSA pipeline. **b.** Schematic of attractor reconstruction with delay embeddings. **c.** Schematic of the Koopman Operator linearization over time-delay coordinates. **d,e.** PCA trajectories (black lines) of a GRU (left) and Vanilla RNN (right) trained to produce sine waves of a range of frequencies (Sussillo and Barak [2013]). Colored dots indicate center points of the limit cycles, colored by frequency of the output wave. **f.** Comparison of measured dissimilarity between the networks in **d,e** with (from left to right) Procrustes on the condition-averaged trajectories, Procrustes on the delay-embedded condition-averaged trajectories, DSA on the original data, and DSA on the delay embedding. Lower is more similar. Each metric uses angular distance, which ranges from 0 to π.

Procrustes Analysis over Vector Fields Procrustes Analysis is a shape metric which compares the geometric similarity of two datasets after identifying an optimal orthogonal transformation to align them (Dryden and Mardia [2016]). To use it in DSA, we must modify the notion of orthogonal transformations to be compatible with linear systems. Recall that the Procrustes metric solves the following minimization problem (Williams et al. [2021]):

$$\mathrm{d}(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{C}\in O(n)} ||\mathbf{X} - \mathbf{C}\mathbf{Y}||_F \tag{4}$$

Where $O(n)$ is the orthogonal group ($C^TC = I$), F indicates the Frobenius norm, and X, Y are two data matrices. A distance metric that is invariant to orthogonal transformations captures the intuitive notion that there is no privileged basis across neural networks.

However, vector fields, represented by the DMD matrices $\mathbf{A}_x, \mathbf{A}_y$ above, transform differently than vectors X and Y (see Supplementary Information Section 8 for an illustration). The invariance we seek is that of a conjugacy, or change of basis, between linear systems, which is reflected in the similarity transform, CA_xC^{-1}. Note that this is a special case of equation 1 over linear systems. Thus we introduce a novel similarity metric, which we call *Procrustes Analysis over Vector Fields*:

$$\mathrm{d}(\mathbf{A}_x, \mathbf{A}_y) = \min_{\mathbf{C}\in O(n)} ||\mathbf{A}_x - \mathbf{C}\mathbf{A}_y\mathbf{C}^{-1}||_F \tag{5}$$

https://doi.org/10.52202/075280-1467

The angular form of this distance can be written as:

$$\mathrm{d}(\mathbf{A}_x, \mathbf{A}_y) = \min_{\mathbf{C}\in O(n)} \arccos \frac{\mathbf{A}_x \cdot (\mathbf{C}\mathbf{A}_y\mathbf{C}^{-1})}{|\mathbf{A}_x|_F \, |\mathbf{A}_y|_F} \tag{6}$$

where $\mathbf{A}_x \cdot (\mathbf{C}\mathbf{A}_y\mathbf{C}^{-1})$ is computed as $\mathrm{Tr}(\mathbf{A}_x^T \mathbf{C}\mathbf{A}_y\mathbf{C}^{-1})$. In the Supplementary Information Section 9, we prove an equivalence between DSA and the 2-Wasserstein distance between the two DMD matrices' eigenspectrum, in special cases. This connection highlights the grounding of DSA in Koopman operator theory to measure conjugacy: when two dynamical systems are conjugate, their Koopman eigenspectra are equivalent (Proof in Supplementary Information 10, Budišić et al. [2012], Lan and Mezić [2013]). We derive equation 5 from first principles and prove that it is indeed a proper metric in the Supplementary Information (6); the proper metric property is crucial for various downstream machine learning pipelines such as clustering or classification (Williams et al. [2021]).

Equations 5 and 6 are nonconvex optimization problems similar to those in Jimenez et al. [2013], although we solve them with gradient-based optimization. Across our experiments, we found that the metric asymptotes in roughly 200 iterations using the Adam optimizer (Kingma and Ba [2014]) with a learning rate of 0.01. To optimize over the orthogonal group, $O(n)$, we utilize an extension of the Cayley Map, which parameterizes $SO(n)$ (Supplementary Information 5.1). In our experiments, we compare the response of DSA with direct application of Procrustes to the data matrices as it is the most analogous shape metric. If only the identification of conjugacy is desired, metric properties can be relaxed by optimizing C over the larger group of invertible matrices, $GL(n)$.

3 Results

To demonstrate the viability of our method, we identified four illustrative cases that disentangle geometry and dynamics, and additionally performed an ablation analysis of DSA. First, a set of neural networks that are different by shape analyses but are dynamically very similar; second, a set of neural networks that are similar by shape analyses but are dynamically are very different; third, a ring attractor network whose geometry we can smoothly and substantially deform while preserving the attractor topology; finally, a line attractor network whose topology is transformed into a ring attractor by adding periodic boundary conditions.

3.1 Ablation Study

Importantly, both the reduced-rank regression and delay-embedding in DSA are required to isolate dynamical structure. We demonstrate this in Fig. 1f, where we compare the hidden state trajectories of an RNN and GRU solving the Sine Wave task (Sussillo and Barak [2013]) using the standard Procrustes Analysis (via the netrep package, Williams et al. [2021]), delay-embedded Procrustes Analysis (an ablation), DSA with no delay embedding (another ablation), and DSA score. To be exhaustive, we also computed a similarity metric that compares purely temporal information, defined as the angular Procrustes distance over the power spectrum of the RNN's hidden state trajectories. On the RNNs in Fig. 1, this metric reported 1.41. Only DSA identifies the systems as highly similar, indicating that both the delay embedding and the DMD are necessary to abstract geometric particulars from each system.

3.2 DSA captures dynamical similarity between RNNs despite differences in geometry.

We trained a collection of 50-unit RNNs on the well-known 3-bit Flip Flop task from Sussillo and Barak [2013]. In this task, an RNN is presented with a time series from three independent channels, each sporadically occurring pulses of either -1 or 1. The network's objective is to continuously output the last pulse value for each channel until it is updated by a new pulse (hence the Flip-Flop name). Following Maheswaranathan et al. [2019], We varied architecture (LSTM, GRU, UGRNN, Vanilla RNN), activation function (ReLU, Tanh), learning rate (.01,0.005), and multiple different seeds across each set of hyperparameters, ultimately collecting a dataset of 240 networks. We only saved a network if it solved the task with MSE less than 0.05. After training, we tested all models on the same 64 input sequences and extracted the recurrent states, upon which we apply Procrustes and DSA in parallel to compare networks.

Our RNNs have different geometry but equivalent attractor topology (Maheswaranathan et al. [2019]). Fig.2a. displays the trajectories of individual trials in a sample network in the first three principal

https://doi.org/10.52202/075280-1467

components, which capture on average 95.9% of the total variance. The computational solution to this task is characterized by eight stable fixed points at the vertices of a cube that correspond to the eight possible unique memory states (2^3). Importantly, this cubic structure is preserved across all networks, even though particular geometric aspects of the trajectories may differ. Standard shape analyses differentiate architectures and activation function by geometry, which we replicated in Fig. 2b (architecture) and in the Supplementary Information (activation, 13).

We fit individual HAVOK models with 75 delays and rank 100 to sampled trajectories from each network. For computational efficiency, we reduced the dimensionality of the network to 10 using Principal Component Analysis (PCA) before applying HAVOK. In Supplementary Information 14, we demonstrate that HAVOK's predictivity is robust to hyperparameter variation. Because the variance explained from the later PCs is negligible, they can be disregarded while preserving the original dynamics. This improves HAVOK's fit to the data and reduces the memory and time complexity of DSA. For each pair of networks, we compared these states directly using Procrustes Analysis and on the DMD matrices using Procrustes Analysis on Vector Fields. This yields two dissimilarity matrices (DMs), which we reduced to two dimensions using Multidimensional Scaling (MDS, Kruskal [1964]). We plot the results in this similarity space in Fig. 2, where each point is a trained network. As expected, the networks cluster by architecture (Fig. 2b) and activation function (Supplementary Information) when Procrustes is applied, indicating that aspects of their geometry differ. However, the networks intermingle under DSA (Fig. 2c, inset zoomed in), and have pairwise dissimilarities close to zero, indicating that their underlying dynamics are equivalent, which is correct (Maheswaranathan et al. [2019]). We quantified these differences by applying a linear SVM to the dissimilarity matrices and plotted the test accuracy in Fig. 2d. This further underscores the differences between Procrustes and DSA in Fig. 2b and c, as DSA's accuracy is close to chance, whereas Procrustes achieves high accuracy on both labels.

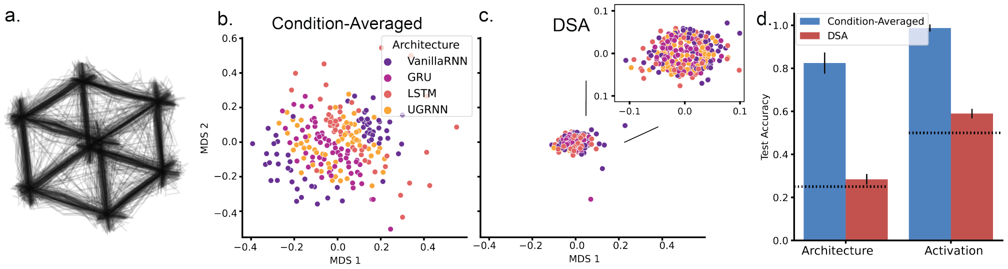

Figure 2: **Same dynamics, different shape, only identified as similar with DSA**. Dots indicate a single trained network. Analysis of RNNs trained on the 3-bit FlipFlop task. **a.** PCA trajectories of a single trained RNN. **b.** MDS projection of the dissimilarity matrix (DM) computed across condition-averaged hidden states between each RNN (architecture indicated by color). In this view of the data, RNNs cluster by architecture. **c.** MDS embedding of the DM generated from DSA of the same set of RNNs as in b. Here, RNNs do not cluster by architecture. **d.** Classification test accuracy of condition-averaged Procrustes and DSA similarity spaces on both architecture and activation labels. Dotted lines indicate chance.

3.3 DSA identifies dynamical differences despite geometric similarity

Next, we sought to display the converse, namely that our similarity method can identify differences in dynamical structure even when spatial analysis cannot. To do so, we examined a set of three noisy recurrent networks from Galgali et al. [2023]. These systems implement a bistable pair of attractors, a line attractor, and a single stable fixed point at the origin. These systems describe three strategies in a classical perceptual decision-making task from systems neuroscience: unstable evidence integration, stable integration, and leaky integration. Their dynamical equations are included in the Supplementary Information. Here, the inputs to the latter two networks were adversarially optimized so that the condition-averaged trajectories are indistinguishable from the first network (Fig. 3a). These networks have intrinsic noise in their hidden states, which enables DSA to distinguish systems beyond condition averages. We simulated 200 noisy trajectories from 100 networks per class, each with randomly sampled parameters.

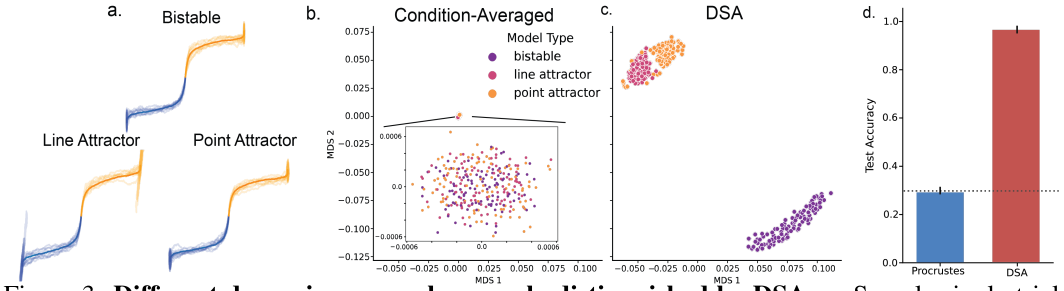

Figure 3: **Different dynamics, same shape, only distinguished by DSA. a.** Sample single-trial (faded) and condition-averaged trajectories of the three systems in two dimensions (x and y axes). Trials start at the center point and move away in a direction depending on the condition (colored). **b.** Procrustes-MDS on the dissimilarity matrices of the three network types. **c.** DSA-MDS of the three networks. **d.** Classification test accuracy of condition-averaged Procrustes and DSA similarity spaces. Dotted line indicates chance.

To confirm that the condition-averaged trajectories are indistinguishable, we visualize the condition-averaged and single-trial trajectories in Fig. 3a for each system. We fit HAVOK models with 100 lags and a rank of 50 for each network and computed DSA and Procrustes pairwise (5,000 total comparisons). As before, we plot a low-dimensional MDS projection of these dissimilarity matrices (3). When the two types of similarities are compared side by side (spatial similarity, Fig. 3b, and DSA, Fig. 3c), it is evident that only our method is capable of easily distinguishing dynamic types from the data alone. This is quantified by almost perfect test accuracy via a linear classifier in Fig. 3d. Thus, DSA can efficiently extract features that describe the underlying dynamical system from data alone, which better reflect the nature of the computation that is being performed in RNNs.

3.4 DSA is invariant under geometric deformations that preserve attractor topology

Next, we examined how DSA responds when smooth geometric deformations are applied to a dynamical system. Here, we chose to study a ring attractor network, a system known to track head direction in the brain, and whose topology is a ring embedded in n neurons (Zhang [1996], Chaudhuri et al. [2019], Skaggs et al. [1995]). The neural activity on the ring is instantiated as a bump of local activity, which is stable due to the localized and translationally-symmetric connectivity profile. In each trial, we randomly sampled n uniformly between 100 and 250, and drove the network with constant input and dynamical noise, leading the bump to rotate around the ring at a roughly constant velocity (we used code from Wang and Kang [2022]). The network's dynamics are detailed in the Supplementary Information (16). Importantly, after simulation, the saved activations can be transformed by a continuous deformation that preserves the ring topology while changing the geometry:

$$r = \frac{1}{1 + \exp\left(-\beta s\right)} \tag{7}$$

The variable s describes the synaptic activations of the ring network (the dynamical variable), and r describes the neural activations, which we compare with DSA and Procrustes. Here, β controls the width of the bump. As β increases, the width of the bump becomes progressively tighter, which makes the ring less planar (see Fig. 1 in Kriegeskorte and Wei [2021] for a depiction). We scaled the magnitude of β from 0.1 to 4. In the inset of Fig. 4b, we quantified the ring's dimensionality across β, measured as the participation ratio of the principal components' eigenvalues: $PR = \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2}$ (Gao et al. [2017]).

After applying these transformations to simulated trajectories from our network, we applied DSA and Procrustes to respectively compare all βs to the lowest β in our dataset. In Fig. 4a we visualize the hidden state trajectory in the first two PCs as the ring is progressively transformed with β. We fit HAVOK models with a lag of 10 and a rank of 1000 to trajectories at each level of deformation. When we compared the network using DSA and Procrustes, we found that only DSA is invariant across these transformations (Fig. 4b). While the value for DSA is not strictly zero, this may be due to approximation or numerical error in HAVOK and the similarity metric, as it remains close to zero

https://doi.org/10.52202/075280-1467

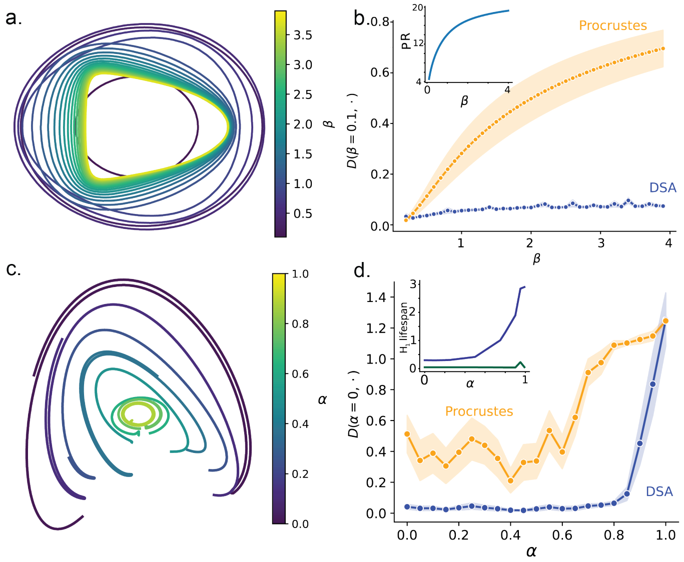

Figure 4: **DSA is invariant under geometric deformation of a ring attractor, and sensitive to the transformation of a ring attractor into a line.** Top row displays scaling of β in the sigmoid, making the ring progressively less planar. The bottom row displays topological transformation of a ring into a line via α. The network is a curve line attractor at $\alpha < 1$, and is a ring attractor at $\alpha = 1$. **a, c**. Trajectories along the ring plotted in the top two PCs of the network, colored by magnitude of the deformation parameter. In **c**, trajectories are scaled for visualization purposes. **b,d**. Distance (D) to $\beta = 0.1$, and $\alpha = 0.0$ in both Procrustes and DSA. Importantly, the topology only changes in the last row, at $\alpha = 1.0$. Inset **b**: Participation Ratio of the ring, a natural continuous measure of dimensionality based on explained variance ratios of PCA eigenvalues (Gao et al. [2017]). Inset **d**. Persistent Homology Analysis of the ring: top 2 lifespans of Betti number 1 features, indicating that the data becomes more ring-like as α increases.

throughout (between 0.05 and 0.12). This suggests that DSA identifies similarity at the level of the topology of a dynamical system.

3.5 DSA responds to changes in topology

Finally, we assessed the converse to the previous analysis: does DSA respond to changes in topology? Using the same ring attractor network, we continuously varied the strength of its periodic boundary conditions to convert the ring into a curved line segment attractor by ablating some neurons from the network on one end of the ring. Our ring is specified by a cosine kernel that defines how each neuron connects to its neighbors. This kernel has a length parameter l which specifies how far apart two neurons can connect. To completely eliminate the boundary conditions, l neurons must therefore be ablated. Using an ablation length c as the number of neurons to ablate in the ring, we defined the following parameter to normalize the transformation across different network sizes.

$$\alpha = 1 - \frac{c}{l} \tag{8}$$

When $\alpha = 1$, there is no ablation and the network is perfectly a ring. When $\alpha < 1$, the network breaks into a line segment attractor. As $\alpha \to 0$ the ring becomes progressively less curved. To ensure that network sizes were consistent across all values of α, we initialized the network size to be $n + c$ before ablation, where n is the desired final number of neurons. We simulated networks of sizes between 200 and 300 neurons. Instead of a constant drive of input as before, we drove the network with a fixed magnitude input u whose sign flipped stochastically. We chose this input to prevent the bump of activity from getting trapped on one end of the line attractors.

When we compared each value of α of the network to $\alpha = 0$ with DSA and Procrustes, we identified that DSA reports values close to 0 until α becomes close to 1.0, when it jumps almost to $\pi/2$. We plotted these results in Fig. 4, where panel **c** depicts the low-dimensional visualization of the line and the ring, and **d** displays our metric's response. Note that because the data are noisy, we expected DSA

https://doi.org/10.52202/075280-1467

to jump slightly before 1.0, which we empirically verified to be around $\alpha = 0.9$ in our simulations. Here we used only 10 delays (for a total embedding size between 2000 and 3000) and rank 100, but found that our results generalized to much larger DMD models as well. This further solidifies the notion that DSA captures topological similarity of two dynamical systems.

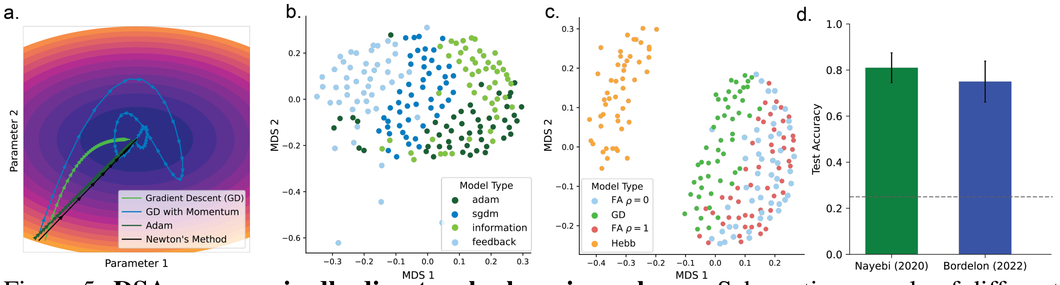

Figure 5: **DSA unsupervisedly disentangles learning rules. a.** Schematic example of different algorithms minimizing $3x^2 + 2y^2$. **b.** DSA MDS plot of hidden activation observables from networks trained on ImageNet **c.** DSA MDS plot of the dynamics of learning in 3-layer linear networks. **d.** Test accuracy of a linear decoder on the DSA similarity space for the model classes. Bars indicate standard deviation over a wide range of DSA hyperparameters.

3.6 DSA can disentangle learning rules in an unsupervised fashion

It is still unclear what learning rules the brain uses. Recent progress includes the design of novel biologically-plausible learning rules (Lillicrap et al. [2016], Hinton [2022]), experiments and modeling work that suggests signs of gradient-based learning in the brain (Sadtler et al. [2014], Humphreys et al. [2022], Payeur et al. [2023]), supervised methods to classify learning rules from aggregate statistics in neural data (Nayebi et al. [2020]), and theory describing how representations evolve across training in different learning rules (Bordelon and Pehlevan [2022]). Because artificial networks are initialized with random seeds, their trajectories in the activity space across learning will almost always have different geometry, despite the fact that they may converge to the same minimum. However, because the trajectories arise from the same dynamical system, DSA should be able to capture similarities between learning trajectories from different initializations. We visualize a variety of learning rule trajectories in Fig. 5a to demonstrate how qualitatively different their dynamics can be. We applied DSA on two test cases from the literature to demonstrate that we can empirically disentangle artificial networks trained with different learning rules. To do so, we fit DSA to trajectories of the neural representations (or observables) across training epochs.

In the first case, we drew 21 aggregate statistics from the dataset from Nayebi et al. [2020], which describe the evolution of synaptic weights and activations of layers in various convolutional neural networks trained on multiple tasks with four learning rules. The learning rules utilized in the dataset include Adam (Kingma and Ba [2014]), Stochastic Gradient Descent with Momentum (Sutskever et al. [2013]), Information Alignment (Kunin et al. [2020]), and Feedback Alignment (Lillicrap et al. [2016]). For each analysis, we fixed one task and generated 50 datasets by randomly selecting (without replacement) subsets of trajectories for each learning rule. On each dataset, we fit a HAVOK model with a rank of 45 and lag of 4 to trajectories of these statistics *across* training. We compared each model pairwise, which results in a dissimilarity matrix. After comparing trajectories of these observables across training with DSA, we visualize the MDS of the similarity space in Fig. 5b. The training dynamics clearly cluster by learning rule under DSA. Quantifying the degree of clustering, we could classify the learning rule with 88.5% test accuracy using only a *linear* classifier and 5-fold cross-validation, whereas the linear SVM classifier in Nayebi et al. [2020] only reached $\sim 55\%$ accuracy (Fig. 2a. in their paper). The test accuracy over a wide range of DSA hyperparameters was 81 % (Fig. 5d).

In the second case, we trained a set of small 3-layer feedforward neural networks (with a hidden size of 100 units) trained to solve a multivariate regression task from Bordelon and Pehlevan [2022]. The learning rules in the second set include Gradient Descent, two forms of Feedback Alignment that vary based on the initialization of the random feedback weights (Boopathy and Fiete [2022]), and Hebbian learning (Hebb [1949]). For each learning rule, we trained 50 networks on 1000 epochs with random initializations and randomly generated data. We fit HAVOK models with a delay of 100 and a rank of 500 onto individual networks' activations across learning. As above, we found that

https://doi.org/10.52202/075280-1467

the MDS visualization of these networks clustered by learning rule in the pairwise similarity space (Fig. 5c). The two types of Feedback Alignment cluster together, suggesting that their dynamics are similar, as might be expected. Once again utilizing a linear classifier, we achieved 89% test accuracy after assigning both types of FA as the same class, and on average 80% accuracy over a range of hyperparameters (Fig. 5d). These results suggest the DSA could be used in a hypothesis-driven fashion to assess learning rules in data recorded from biological circuits.

4 Discussion

We demonstrated that our novel metric, DSA, can be used to study dynamics in neural networks in numerous scenarios, including the comparison of dynamics between different RNNs, dynamics of a single RNN across training, or dynamics of weight and activity changes in feedforward networks over training. We performed an ablation study in Fig. 1f, which demonstrates that both delay embedding and the DMD were required for DSA to succeed. While we applied our metric only to RNNs, we stress that it can be applied to any dynamical system, including diffusion models or generative transformers. Procrustes Analysis over Vector Fields could also be applied to Koopman Embeddings identified via other algorithms, such as an autoencoder (Lusch et al. [2018]).

As suggested by Figs.2, 3, 4 and our theoretical grounding (9, 10), DSA can capture similarity at the level of topology of the dynamics, also known as conjugacy. This can also be recognized from the methodology of DSA: the similarity transform is a conjugacy between two linear systems. In recent work, Redman et al. [2023] used the 2-Wasserstein Distance over the Koopman eigenspectrum to compare learning dynamics of fully-connected networks trained to classify handwritten digits. This metric is quite analogous to ours in its theoretical motivation. In Supplementary Information 9 we prove that their metric is a specific case of DSA. Thus, DSA quantifies how far two dynamical systems are from being conjugate.

As we demonstrate in Fig. 3, our method could be used to quantitatively compare various hypotheses about the dynamics of a particular biological neural circuit (such as fixed point structure), by calculating similarities of various RNNs or bespoke models to data (Pagan et al. [2022], Schaeffer et al. [2020]). This task is typically computationally intensive, relying on assessing the dynamics of noise perturbations in the neural state space (Chaudhuri et al. [2019], Galgali et al. [2023]) or computing the locations of fixed points in a task-optimized RNN model that has similar spatial trajectories as the neural data (Mante et al. [2013], Chaisangmongkon et al. [2017]). Our method will be useful in the analysis of biological neural systems, where only recorded neural activity is accessible. To our knowledge, DSA is the first that can be used to quantitatively assess different dynamical hypotheses. We suggest that this method should be used in conjunction with shape analyses when assessing fits of computational models to data. In future work, we plan to apply DSA to experimental data.

DSA could potentially be used for rapid alignment of Brain-Computer Interfaces (BCI) across days in a subject. Non-stationary sampling of neurons from intracortical electrode shift, neuron death, and other noise plague longitudinal BCI trials, requiring the subject to relearn BCI control daily in a slow and tedious process (Biran et al. [2005], Perge et al. [2013]). Two recent approaches (Willett et al. [2023], Degenhart et al. [2020]) utilized Procrustes Analysis in their decoding and alignment pipeline. As demonstrated in Figs. 2, 3, and 4, an alignment method that takes dynamics into account will be useful. Karpowicz et al. [2022] utilized LFADS to align dynamics across days, but DSA may provide equal performance with substantially higher data efficiency and less compute due to the method's simplicity.

In Section 3.6 we demonstrated how the DSA metric space could be used in a hypothesis-driven fashion to compare neurophysiological data recorded across training to artificial neural networks trained with different learning rules. To do so, one would train a variety of neural networks with different learning rules to solve the same task as the animal model. By applying DSA pairwise to each network and the neurophysiology data, and identifying which learning rules cluster to the experimental data, we can reject other learning rule hypotheses. DSA has multiple advantages over previous methods for this problem: Unsupervised feature generation can improve classification capabilities, and abstraction from geometrymakes the method more generically applicable across systems of different types and initial conditions.

https://doi.org/10.52202/075280-1467

Limitations It can be challenging to identify the optimal DMD hyperparameters, but it is not computationally intensive to perform a grid sweep. Additionally, we found that the ranks and delays required to achieve our results were relatively small. On a V100 GPU, a single DSA took between 30 seconds and a few minutes, depending on the hyperparameters and size of the dataset.

Acknowledgments The authors are grateful to Antonio Carlos Costa for bringing Koopman operators to our attention, and to Sarthak Chandra, Raymond Wang, Aran Nayebi, Ho Kyung Sung, Will Redman, Tosif Ahamed, and members of the Fiete lab for helpful discussions and advice.

References

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv*, 2020. URL `https://www.biorxiv.org/content/early/2020/01/02/407007`.

Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature Neuroscience*, 22(9):1512–1520, 2019.

Alan D. Degenhart, William E. Bishop, Emily R. Oby, Elizabeth C. Tyler-Kabara, Steven M. Chase, Aaron P. Batista, and Byron M. Yu. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature Biomedical Engineering*, pages 672–685, 2020.

Marino Pagan, Vincent D Tang, Mikio C. Aoi, Jonathan W. Pillow, Valerio Mante, David Sussillo, and Carlos D. Brody. A new theoretical framework jointly explains behavioral and neural variability across subjects performing flexible decision-making. *bioRxiv*, 2022. URL `https://www.biorxiv.org/content/early/2022/11/28/2022.11.28.518207`.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, 2008.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. In *Advances in Neural Information Processing Systems*, volume 34, pages 4738–4750, 2021.

Lyndon R Duong, Jingyang Zhou, Josue Nassar, Jules Berman, Jeroen Olieslagers, and Alex H Williams. Representational dissimilarity metric spaces for stochastic neural networks. *arXiv preprint arXiv:2211.11665*, 2022.

Shun-Ichi Amari and Michael A Arbib. Competition and Cooperation in Neural Nets. *Systems Neuroscience, Academic Press, J. Metzler (ed).*, pages 119–165, 1977.

J J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U. S. A.*, 81(10):3088–3092, 1984.

J J Hopfield and D W Tank. Computing with neural circuits: a model. *Science*, 233(4764):625–633, 1986.

H S Seung. How the brain keeps the eyes still. *Proc. Natl. Acad. Sci. U. S. A.*, 93(23):13339–13344, 1996.

K Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *J. Neurosci.*, 15:2112–2126, 1996.

https://doi.org/10.52202/075280-1467

Richard H. R. Hahnloser, Rahul Sarpeshkar, Misha A. Mahowald, Rodney J. Douglas, and H. Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 2000.

Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput. Biol.*, 5(2):e1000291, February 2009.

Xiao-Jing Wang. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, 2002.

David Sussillo and L F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487 (7405):51–56, 2012.

Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

Mikail Khona and Ila R. Fiete. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 23(12):744–766, 2022.

Aniruddh R. Galgali, Maneesh Sahani, and Valerio Mante. Residual dynamics resolves recurrent contributions to neural computation. *Nature Neuroscience*, 2023.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

Gregory Snyder and Zhuoyuan Song. Koopman operator theory for nonlinear dynamic modeling using dynamic mode decomposition. *arXiv preprint arXiv:2110.08442*, 2021.

Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern Koopman Theory for Dynamical Systems, 2021. URL `http://arxiv.org/abs/2102.12086`.

KiJung Yoon, Michael A. Buice, Caswell Barry, Robin Hayman, Neil Burgess, and Ila R. Fiete. Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nature neuroscience*, 16(8):1077–1084, 2013.

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

Bingni W. Brunton, Lise A. Johnson, Jeffrey G. Ojemann, and J. Nathan Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, pages 1–15, 2016a.

Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, and J. Nathan Kutz. Chaos as an Intermittently Forced Linear System. *Nature Communications*, 8(1):19, 2017.

Pierre Dubois, Thomas Gomez, Laurent Planckaert, and Laurent Perret. Data-driven predictions of the Lorenz system. *Physica D: Nonlinear Phenomena*, 408:132495, 2020.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016b.

Hassan Arbabi and Igor Mezić. Ergodic Theory, Dynamic Mode Decomposition, and Computation of Spectral Properties of the Koopman Operator. *SIAM Journal on Applied Dynamical Systems*, 16 (4):2096–2126, January 2017.

https://doi.org/10.52202/075280-1467

Mason Kamb, Eurika Kaiser, Steven L. Brunton, and J. Nathan Kutz. Time-Delay Observables for Koopman: Theory and Applications. *SIAM Journal on Applied Dynamical Systems*, 19(2): 886–917, 2020.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence*. Warwick, 1981.

David Sussillo and Omri Barak. Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks. *Neural Computation*, 25(3):626–649, 2013.

Ian Dryden and Kanti Mardia. *Statistical shape analysis, with applications in R: Second edition*. 2016.

Marko Budišić, Ryan Mohr, and Igor Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, December 2012.

Yueheng Lan and Igor Mezić. Linearization in the large of nonlinear systems and koopman operator spectrum. *Physica D: Nonlinear Phenomena*, 2013.

Nicolas D. Jimenez, Bijan Afsari, and Rene Vidal. Fast Jacobi-type algorithm for computing distances between linear dynamical systems. In *2013 European Control Conference (ECC)*, pages 3682–3687, Zurich, 2013. IEEE.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

W. E. Skaggs, J. J. Knierim, H. S. Kudrimoti, and B. L. McNaughton. A model of the neural basis of the rat's sense of direction. *Advances in Neural Information Processing Systems*, 7:173–180, 1995.

Raymond Wang and Louis Kang. Multiple bumps can enhance robustness to noise in continuous attractor networks. *PLOS Computational Biology*, 18(10):1–38, 10 2022.

Nikolaus Kriegeskorte and Xue-Xin Wei. Neural tuning and representational geometry. *Nature Reviews Neuroscience*, 22(11):703–718, 2021.

Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, 2017. URL `https://www.biorxiv.org/content/early/2017/11/12/214262`.

Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7(1): 13276, 2016.

Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

Patrick T. Sadtler, Kristin M. Quick, Matthew D. Golub, Steven M. Chase, Stephen I. Ryu, Elizabeth C. Tyler-Kabara, Byron M. Yu, and Aaron P. Batista. Neural constraints on learning. *Nature*, 512 (7515):423–426, 2014.

Peter C. Humphreys, Kayvon Daie, Karel Svoboda, Matthew Botvinick, and Timothy P. Lillicrap. Bci learning phenomena can be explained by gradient-based optimization. *bioRxiv*, 2022. URL `https://www.biorxiv.org/content/early/2022/12/08/2022.12.08.519453`.

Alexandre Payeur, Amy L. Orsborn, and Guillaume Lajoie. Neural manifolds and gradient-based adaptation in neural-interface tasks. *bioRxiv*, 2023. URL `https://www.biorxiv.org/content/early/2023/03/12/2023.03.11.532146`.

https://doi.org/10.52202/075280-1467

Aran Nayebi, Sanjana Srivastava, Surya Ganguli, and Daniel L Yamins. Identifying Learning Rules From Neural Network Observables. In *Advances in Neural Information Processing Systems*, volume 33, pages 2639–2650, 2020.

Blake Bordelon and Cengiz Pehlevan. The Influence of Learning Rule on Representation Dynamics in Wide Neural Networks, 2022. URL `http://arxiv.org/abs/2210.02157`.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1139–1147. PMLR, 2013.

Daniel Kunin, Aran Nayebi, Javier Sagastuy-Brena, Surya Ganguli, Jonathan Bloom, and Daniel Yamins. Two routes to scalable credit assignment without weight symmetry. In *International Conference on Machine Learning*, pages 5511–5521. PMLR, 2020.

Akhilan Boopathy and Ila Fiete. How to Train Your Wide Neural Network Without Backprop: An Input-Weight Alignment Perspective. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, 1949.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1), 2018.

William T. Redman, Juan M. Bello-Rivas, Maria Fonoberova, Ryan Mohr, Ioannis G. Kevrekidis, and Igor Mezić. On equivalent optimization of machine learning methods, 2023.

Rylan Schaeffer, Mikail Khona, Leenoy Meshulam, Brain Laboratory International, and Ila Fiete. Reverse-engineering recurrent neural network solutions to a hierarchical inference task for mice. In *Advances in Neural Information Processing Systems*, 2020.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 2013.

Warasinee Chaisangmongkon, Sruthi K. Swaminathan, David J. Freedman, and Xiao-Jing Wang. Computing by robust transience: How the fronto-parietal network performs sequential, category-based decisions. *Neuron*, 2017.

Roy Biran, David C. Martin, and Patrick A. Tresco. Neuronal cell loss accompanies the brain tissue response to chronically implanted silicon microelectrode arrays. *Experimental Neurology*, 2005.

János A. Perge, Mark L. Homer, Wasim Q. Malik, Sydney Cash, Emad Eskandar, Gerhard Friehs, John P. Donoghue, and Leigh R. Hochberg. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *Journal of Neural Engineering*, 2013.

Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. preprint, Neuroscience, 2023. URL `http://biorxiv.org/lookup/doi/10.1101/2023.01.21.524489`.

Brianna M. Karpowicz, Yahia H. Ali, Lahiru N. Wimalasena, Andrew R. Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Lee E. Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *bioRxiv*, 2022. URL `https://www.biorxiv.org/content/early/2022/11/08/2022.04.06.487388`.

https://doi.org/10.52202/075280-1467