Biologically-inspired Deep-learning Workshop

Laura Bernáez Timón[1], Angus Chadwick[2], Isabel Cornacchia[2], Maximilian F. Eggl[3], Arthur Pellegrino[2], Janko Petkovic[4]

Abstract

A specialized workshop, the third in a series of workshops, for young researchers took place in the spring of 2025 in Guntersblum. This was funded by the Joachim Herz Stiftung with the aim to support the design and implementation of interdisciplinary events for young scientists as a continuation of the "Begegnungszonen" program. This program aimed to **support the design and implementation of interdisciplinary events for young scientists**, which perfectly aligns with the intended goals of the workshop series. By providing 16 doctoral students and postdoctoral researchers the opportunity to be exposed to state-of-the-art methods at the cutting edge of machine learning and life sciences, an interdisciplinary environment was created that allowed them to learn things beyond the fields they normally work in. The speaker for the 2025 workshop on the topic of the Dimensionality reduction techniques was Dr. Angus Chadwick of Institute for Adaptive and Neural Computation at the School of Informatics, University of Edinburgh. He was assisted by his doctoral student, Isabel Cornacchia, and postdoctoral fellow, Arthur Pellegrino. This report briefly introduces the workshop series' underlying idea, a motivation for the study and the importance and impact of Simulation-based inference, and a representative exposé of the students' work during the workshop.

1 The Workshop format

Exchanges between established researchers and early-career scientists are vital to scientific progress. They allow new ideas to circulate throughout the community while creating a network of support for emerging talent. To foster this exchange and enable interdisciplinary networking between early-career researchers and recognized experts, a workshop was held in Guntersblum, Germany, supported by the Joachim Herz Foundation.

This third edition of the "Bio-inspired Deep Learning" series introduced a format that is still relatively uncommon in the neuroscience community. It was deliberately chosen to strengthen interactions across career stages and provide participants with exposure to cutting-edge research at the intersection of life sciences, computational modeling, and machine learning. First introduced in the two earlier editions of the series, this format is based on a short timeframe (three days) and encourages hands-on learning akin to on-the-job training. Rather than relying on traditional lectures alone, participants work through small, focused projects that expose them to advanced concepts in a pedagogical, applied way. By organizing students into small groups based on familiarity with the topic, experience, and research background, the format fosters a collegial environment that encourages peer learning and scientific exchange.

Crucially, the format is designed to be fundamentally interdisciplinary. Interdisciplinarity is becoming increasingly important in an interconnected science world, and thus, by bringing together scientists from a variety of different fields, the workshop provides a robust exchange of ideas and technologies from different scientific disciplines that enables the participants to become better scientists.

[1]Center for Future Generations
[2]University of Edinburgh
[3]Instituto de Neurociencias, CSIC-UMH
[4]Universitätsklinikum Bonn

In contrast, many current neuroscience workshops follow a more traditional structure: extended timeframes, multiple invited speakers, and a lecture-based format. While effective in many respects, these settings often limit direct interaction between junior researchers and senior scientists. Our workshop series was designed to address this gap, offering a more interactive, mentorship-oriented approach to education in the neuroAI space.

The format of this workshop is directly inspired by the Young ERCOFTAC workshop series, an initiative of Peter Schmid, Shervin Bagheri and Taraneh Sayadi, and which M. Eggl had the pleasure of attending during his PhD. This ongoing workshop series provides a platform for early-career researchers to be exposed to state-of-the-art research within the fluid dynamics community.

Mirroring their approach, 16 PhD students and Postdoctoral Fellows were invited to the Weingut Domhof, a small remote location 30 km south of Mainz. Now operating as a hotel, the Domhof retains its winemaking tradition while offering a quiet, rural environment conducive to focused work. Its remote setting fostered active engagement in daily activities, including communal meals, which contributed to rapid and lasting bonds among participants—connections that often endure beyond the workshop itself.

The scientific program format was then devised to introduce the students (who had very different backgrounds and were at various stages of their scientific careers) to the workshop topic and prepare them for the group work in the days ahead. After an initial set of introductory talks on dynamical systems and dimensionality techniques (I. Cornacchia and A. Pellegrino), the organizing committee (L. Bernaez Timon, J. Petkovic and M. Eggl) and A. Chadwick presented each group with their research topic, which covered a different aspect of dimensionality reduction. The projects were designed to require differing expertise and their scope was confined to tasks that could be feasibly accomplished solely using laptops. While working in their groups, students were closely supervised by their assigned mentor (one of the assistants of A. Chadwick, I. Cornacchia and A. Pellegrino). The direct involvement in a time-limited project and the immediate access to guidance and expertise yielded a more long-lasting result than a passive lecture series. Although each student group concentrated on a specific problem, they all benefited from exposure to the other projects during the final presentation of the results; in this manner, the students were not only proficient in their own project but knowledgeable in related topics.

Thus, with this format in mind, the goals of our workshop were as follows: *i)* expose students to the ever-growing field of dimensionality reduction and dynamical systems, *ii)* foster networking between the students by arranging them in groups and giving them state-of-the-art research projects to complete and *iii)* present the participants an opportunity to meet and discuss with experts in the field.

After receiving feedback from all iterations of this workshop, we are confident we have achieved, and continue to achieve, all these goals. In future iterations, this workshop can establish itself as a significant opportunity for young scientists to study problems at the intersection of neuroscience and machine learning. By providing specialized training opportunities and by encouraging the formation of collegial networks and connections, we hope to encourage, educate and foster the next generation of scientists in these interesting and important fields.

2 Topic of the workshop: Dimensionality reduction techniques.

The behaviour of a wide array of systems, both biological and not, can be effectively described using the theory of dynamical systems. For example, neural circuits in neuroscience (either on single neuron

level or the collective dynamics of neuronal populations) are frequently studied through a dynamical systems lens, offering a powerful framework to capture how neural states evolve over time (Vogels et al., 2005). However, these systems and the data associated with them are often high-dimensional, reflecting the large number of interacting components. Reducing the dimensionality of these features to obtain the characteristics that contain the most information and insight is a crucial component in analysing these systems. Yet, in practice, the collective dynamics tend to evolve on constrained, lower-dimensional structures within the high-dimensional state space (Fefferman et al., 2016; Gallego, 2025). When such data arise from an underlying dynamical system, the geometry of the low-dimensional manifold must, in turn, reflect properties of the system's governing equations.

These neural manifolds—low-dimensional surfaces embedded within the high-dimensional neural activity space—capture the dominant modes of coordinated activity across populations of neurons. Instead of behaving independently, neurons tend to co-activate along structured patterns that reflect shared inputs, local connectivity, and task demands. As neural activity unfolds in time, the system traces trajectories on these manifolds, whose shape and dynamics encode essential features of perception, decision-making, and motor control. The existence of neural manifolds suggests that, despite the apparent complexity of the brain's high-dimensional activity, its computations may be governed by relatively low-dimensional latent variables. This insight has motivated a broad array of analytical approaches, where uncovering the structure of these manifolds becomes key to understanding how the brain organizes and processes information (Mitchell-Heggs et al., 2023).

More broadly, the study of neural manifolds offers a unifying framework to interpret motor control and cognitive functions. The geometry and dynamics of these manifolds reflect the connectivity and constraints within neural circuits, shaping both what activity patterns can be generated and what behaviors can be learned (Sadtler et al., 2014). Numerous studies have employed both linear (Smith and Brown, 2003) and nonlinear (Nieh et al., 2021) dimensionality reduction methods to extract these structures and identify latent neural modes (Cunningham and Yu, 2014). These findings reinforce the view that neural computation is deeply shaped by the interplay between high-dimensional dynamics and low-dimensional structure, a hallmark of complex biological systems.

As a leading expert in this field, Dr. Chadwick was the natural choice to lead this year's iteration of the "Bio-inspired deep learning" workshop (Pellegrino et al., 2023, 2024; Pellegrino and Chadwick, 2024; Chadwick et al., 2023). Taking advantage of larger datasets at ever-higher fidelity, Dr. Chadwick with his team has introduced the Manifold Discovery via Dynamical Systems framework. This method allows for simultaneous inference of the low-dimensional manifold and the governing dynamics that produce observed neural activity. Moreover, this approach provides a rigorous, data-driven route toward uncovering the computational principles embedded in neural systems, by for example, linking the heterogeneous responses of individual neurons to the coherent, population-level dynamics responsible for timing flexibility.

Given these expertise, four projects were presented to the student groups:

- **The topology of walking**: Using a idealised model of fly gait dynamics, this project aims to study the topological structure of the underlying manifold under different embedding conditions and to assess how variation in walking speed influences the embedded dynamics. Using this approach, this project then tries to answer the question how the latent structure of leg dynamics allows for stable and adaptive gaits, importantly uncovering the principles of locomotor control.

- **Applying MDDS to a time tracking task**: The goal of this project was to apply MDDS, along with PCA, t-SNE, and temporal generalization analysis, to uncover low-dimensional struc-

tures and control strategies underlying population activity. By comparing the different approaches, the project aims to understand how the brain flexibly regulates time.

- **Applying MDDS to a hand-reaching task**: Similar to the project above, this group aimed to apply different dimensionality reduction techniques (including, PCA, MDDS and UMAP) to neural recordings from primary motor cortex (M1) of a monkey performing a center-out reach task to eight target directions using a joystick, ultimately aiming to understand how neural population activity in the motor cortex encodes movement.
- **Linear manifold-generating dynamical systems**: This project aims to study linear manifold-generating dynamical systems, a class of dynamical systems whose trajectories are constrained to evolve on lower-dimensional submanifolds that together partition the phase space into non-overlapping layers. Importantly, this project wanted to understand what the effect of incorporating nonhomogeneous linear terms under specific algebraic constraints would be.

While all the projects were ambitious in scope and topic, the results achieved by the students are nonetheless impressive and point to interesting future endeavours. We are grateful for all their hard work and openness to work on novel topics.

References

Angus Chadwick, Adil G Khan, Jasper Poort, Antonin Blot, Sonja B Hofer, Thomas D Mrsic-Flogel, and Maneesh Sahani. Learning shapes cortical dynamics to enhance integration of relevant sensory input. *Neuron*, 111(1):106–120, 2023.

John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Juan Alvaro Gallego. Neural manifolds: more than the sum of their neurons. *Nature Reviews Neuroscience*, pages 1–1, 2025.

Rufus Mitchell-Heggs, Seigfred Prado, Giuseppe P Gava, Mary Ann Go, and Simon R Schultz. Neural manifold analysis of brain circuit dynamics in health and disease. *Journal of computational neuroscience*, 51(1):1–21, 2023.

Edward H Nieh, Manuel Schottdorf, Nicolas W Freeman, Ryan J Low, Sam Lewallen, Sue Ann Koay, Lucas Pinto, Jeffrey L Gauthier, Carlos D Brody, and David W Tank. Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595(7865):80–84, 2021.

Arthur Pellegrino and Angus Chadwick. The shared geometry of biological and recurrent neural network dynamics. In *JOURNAL OF COMPUTATIONAL NEUROSCIENCE*, volume 52, pages S17–S18. SPRINGER VAN GODEWIJCKSTRAAT 30, 3311 GZ DORDRECHT, NETHERLANDS, 2024.

Arthur Pellegrino, N Alex Cayco Gajic, and Angus Chadwick. Low tensor rank learning of neural dynamics. *Advances in Neural Information Processing Systems*, 36:11674–11702, 2023.

Arthur Pellegrino, Angus Chadwick, and N Alex Cayco-Gajic. Capturing interpretable dynamics of population-level learning. In *JOURNAL OF COMPUTATIONAL NEUROSCIENCE*, volume 52, pages S28–S29. SPRINGER VAN GODEWIJCKSTRAAT 30, 3311 GZ DORDRECHT, NETHERLANDS, 2024.

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.

Anne C Smith and Emery N Brown. Estimating a state-space model from point process observations. *Neural computation*, 15(5):965–991, 2003.

Tim P Vogels, Kanaka Rajan, and Larry F Abbott. Neural network dynamics. *Annu. Rev. Neurosci.*, 28(1):357–376, 2005.

The topology of walking

Vincent Degenhart[1], Pamela Guruciaga[2], Daniel Panadero Soler[3], Ryo Segawa[4]

Abstract

Understanding coordinated locomotion requires bridging rhythmic motor patterns with the geometry of the underlying dynamics. In this work, we develop a linear dynamical system (LDS) framework that captures key features of locomotor coordination. We begin by analysing the topological effects of delay embedding a minimal LDS, showing that even a single delay can transform planar oscillations into toroidal trajectories, while multiple delays enrich the embedding geometry. We then explore the effect of non-normality, demonstrating how coupling asymmetry introduces directionality and breaks phase symmetry. Extending the model to multi-limb systems, we investigate how specific inter-leg interaction schemes can reproduce biologically relevant gait patterns in hexapods. Finally, we apply delay embedding techniques to these dynamics and show how the resulting geometry encodes the different types of limb coordination. These results illustrate how minimal ingredients—delay, asymmetry, and coupling—can give rise to the rich topology of coordinated motion.

1 Introduction

Coordinated locomotion, such as walking, is a rhythmic behavior arising from structured neural and mechanical dynamics. In insects like Drosophila melanogaster (fruit flies), the legs must move in precisely orchestrated sequences to maintain stable and adaptive gaits. These patterns vary with speed, and understanding the latent structure of such movements is key to uncovering the principles of locomotor control.

Previous work has applied dimensionality reduction techniques to leg position time series in order to characterize the geometry of locomotion. Notably, it was proposed that the state space of Drosophila leg trajectories lies on a cylindrical manifold, as revealed using UMAP (Uniform Manifold Approximation and Projection). This result suggests that the complex coordination of the legs may be driven by relatively simple underlying dynamics.

In this project, we challenge the cylinder interpretation by modeling the system from first principles as a low-dimensional dynamical system. Specifically, we explore how the time evolution of leg positions — particularly when subject to delay embedding — gives rise to geometric structures in state space. Rather than analyzing real recorded data, we work with an idealized model that mimics the core rhythmic nature of leg motion. This allows us to investigate the topological structure of the gait manifold under different embedding conditions and to assess how variation in walking speed influences the embedded dynamics. We hypothesize that the observed trajectories are better understood as lying on a torus, or even a higher-dimensional toroidal structure, depending on the number of delays and degrees of freedom considered.

Background and initial model description

[1]University of Bonn, Bonn, Germany
[2]European Molecular Biology Laboratory, Heidelberg, Germany
[3]Instituto de Neurociencias, Consejo Superior de Investigaciones Científicas - Universidad Miguel Hernández (CSIC-UMH), Sant Joan d'Alacant, Spain
[4]The WAVES research group and the 4Brain group, Ghent University, Ghent, Belgium/imec, Belgium

At the outset of this project, we are provided with a simplified model that abstracts away the raw leg movement data of the fly and instead focuses on capturing its core rhythmic structure. The movement of a single leg, or a pair of legs, is modeled as a planar linear oscillator defined by the differential equation:

$$\dot{x} = \begin{bmatrix} 0 & \pi \\ -\pi & 0 \end{bmatrix} x, \quad x(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{1}$$

This describes circular motion in a 2D state space, where the trajectory lies on a circle — a natural representation for periodic processes. The oscillator mimics the repetitive activation and deactivation of muscles involved in stepping motions.

To uncover the underlying geometry of such periodic signals, a common technique is delay embedding: transforming a one-dimensional or low-dimensional time series into a higher-dimensional space by appending delayed copies of the signal. Formally, a time series $x_i(t) \in \mathbb{R}^N$ is transformed into $y(t_i) = [x(t_i), x(t_i + \Delta t), \ldots, x(t_i + D\Delta t)] \in \mathbb{R}^{N(D+1)}$. This method is rooted in Takens' embedding theorem and is widely used to reconstruct the phase space of dynamical systems from observations.

In the case of the fly leg oscillator, applying a delay embedding with one delay maps the circular trajectory into a torus, $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$, in the higher-dimensional space. Each pair $(\theta(t), \theta(t + \Delta t))$ defines a point on this torus. However, if the oscillator's frequency is fixed, the system's trajectory forms a single loop within the torus. Only when varying the frequency (i.e., walking speed) do we observe a family of such loops that collectively fill out the toroidal structure.

Interestingly, by performing a change of coordinates such as $\hat{\theta}_1 = \theta(t + \Delta t) - \theta(t)$, we can "unroll" or decorrelate the torus into a cylindrical structure — aligning with earlier claims in the literature that the fly's gait manifold resembles a cylinder. This sets the stage for the key question explored in this work: whether this manifold is better understood as a torus or a higher-dimensional extension thereof, depending on how the system is modelled and embedded.

2 Methods

To investigate the underlying structure of periodic motor behaviors, we employ the framework of linear dynamical systems combined with time-delay embedding techniques. Linear dynamical systems offer a tractable way to model oscillatory phenomena, where the system evolves according to a constant matrix acting on the state vector. In the simplest case, this leads to circular trajectories in state space, reflecting the periodic nature of biological movements such as stepping.

To analyze the geometry of these dynamics, we apply delay embedding, a method that reconstructs the system's phase space by stacking time-delayed observations of the original signal. This transforms a one-dimensional or low-dimensional time series into a higher-dimensional trajectory that reveals the underlying structure of the system's attractor.

The delay-embedded space often reveals that the dynamics lie on low-dimensional manifolds such as circles, tori, or higher-dimensional analogues. Access to this manifold enables the use of tools from dynamical systems theory—such as phase analysis, topological classification, and stability analysis—while reducing the complexity of the original high-dimensional system.

This approach is particularly useful in neuroscience and biomechanics, where the observed behavior may arise from complex but low-dimensional latent dynamics. By analyzing these embedded manifolds, we can better understand how rhythmic patterns are structured, coordinated, and modulated.

3 Results

3.1 Topological Effects of Adding Delays

A key question in understanding the structure of periodic motion through delay embedding is how the topology of the system's state space evolves as additional delays are introduced. In our case, the original system is a simple two-dimensional linear oscillator whose state vector evolves on a circle $\mathbb{S}^1$ due to the purely imaginary eigenvalues of the governing linear dynamical system. This implies that the state vector $x(t) \in \mathbb{R}^2$ traces out a periodic trajectory on a one-dimensional circular manifold.

When we apply a time-delay embedding, we extend the state representation to include time-shifted copies of $x(t)$. With one delay ($D = 1$), the embedded state becomes $y(t) = [x(t), x(t + \Delta t)] \in \mathbb{R}^4$. Each of the two vectors $x(t)$ and $x(t + \Delta t)$ lies on a circle, so the embedded trajectory lies on the Cartesian product $\mathbb{S}^1 \times \mathbb{S}^1$, which is topologically a torus. However, if the speed of the oscillator is held constant, then $x(t)$ and $x(t + \Delta t)$ remain tightly correlated, and the system traces a one-dimensional trajectory (still a circle) within the torus.

By introducing velocity variation, the system traverses different orbits within the torus. Each velocity corresponds to a different angular rate, resulting in distinct phase differences between $x(t)$ and $x(t+\Delta t)$. This creates a family of trajectories filling out the toroidal surface, effectively revealing the full toroidal structure of the delay-embedded state space.

Adding a second delay (e.g. $D = 2$) transforms the state space further. The system state now includes $x(t)$, $x(t+\Delta t)$, $x(t+2\Delta t)$, resulting in a six-dimensional vector in $\mathbb{R}^6$ Since each of the three vectors lies on a circle, the state space becomes $\mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{S}^1$, which is a 3-torus or hypertorus. This higher-dimensional structure captures more temporal context of the periodic signal and provides a richer embedding that may reveal subtler features of the underlying dynamics.

However, even in this higher-dimensional toroidal space, varying a single global velocity parameter may not be sufficient to reach every possible point on the hypertorus. This is because all components of the system (each delayed copy of $x(t)$) are still driven by a shared oscillator with the same frequency. Therefore, while the system explores a two-dimensional manifold within the hypertorus (corresponding to combinations of phase and velocity), it does not fill the full three-dimensional space of the hypertorus unless additional independent degrees of freedom—such as independently controlled limb dynamics—are introduced.

Thus, delay embedding increases the dimensionality and richness of the ambient manifold, but the accessible submanifold remains constrained by the number of independent control parameters (in this case, velocity).

3.1.1 Exploring the case of having 2 delays

With only one delay, the delay-embedded system consists of the current state $x(t)$ and a single future state $x(t + \Delta t)$, both in $\mathbb{R}^2$, resulting in a four-dimensional embedding. However, since each vector lies on a circle, the effective topology is a two-dimensional torus. To visualize this, we reduced the representation to two angular variables ($\theta(t)$ and $\theta(t + \Delta t)$), producing a 2D phase plot that reveals how different velocities span the torus.

When a second delay is introduced, the state includes $x(t)$, $x(t + \Delta t)$ and $x(t + 2\Delta t)$, extending the embedding into a six-dimensional space. Using angle representations for each component, this corresponds to a 3-torus. We represent this space through a 3D phase plot using the three angular coordinates, as shown in Fig.1. This visualization clearly illustrates how the accessible states evolve

in a higher-dimensional toroidal geometry and how velocity modulation controls the trajectory within this space.

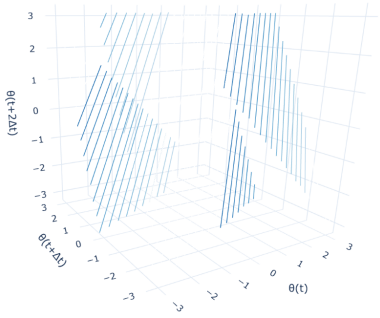


Figure 1: 3D phase plot of the delay-embedded oscillator with two delays, showing the trajectory in angular coordinates $\theta(t)$, $\theta(t + \Delta t)$ and $\theta(t + 2\Delta t)$. Each line corresponds to a different constant velocity.

To better understand how the trajectories span the 3-dimensional torus under variation of velocity, we explored 2D slices of the state space by fixing the value of one of the angular variables, $\theta(t)$. By holding $\theta(t)$ constant and examining the corresponding values of $x(t+\Delta t)$ and $x(t+2\Delta t)$, we projected the high-dimensional manifold onto a 2D plane. We repeated this process across several fixed values of $\theta(t)$ and plotted the resulting points for all considered velocities.

In these 2D projections (Fig.2), we observed that the points formed straight lines with slope equal to the the ratio between the delays' magnitudes. This outcome reflects the fact that for a given velocity and a fixed starting phase, the phase angles at successive time steps evolve linearly, resulting in straight trajectories in angular space.

However, although this slicing illustrates some of the toroidal structure, it also reveals an important limitation: only a subset of the 3-torus is accessible when velocity is the only varying parameter. This is because the oscillator's evolution remains tightly constrained by its initial phase and linear dynamics. While different velocities trace different paths, each remains confined to a 1D curve within the full 3D torus. Therefore, velocity modulation alone is not sufficient to explore the entire hypertorus; full coverage would require independent variation of other degrees of freedom such as phase or coupling parameters.

3.2 Feedforward coupling between different oscillators

By fitting a linear dynamical system to the fly gait data, previous work identified non-normal dynamics, quantified using the Henrici departure from normality measure. This measure, which is defined as

$$0 \leq \frac{||WW^T - W^TW||_F}{2 \cdot ||W||_F^2} \leq 1$$

where $||\cdot||_F$ is the Frobenius norm, revealed significant deviations from normality in the system matrix W, indicating that the eigenvectors are non-orthogonal. The presence of such dynamics suggests that the fly's gait control system may leverage short-term non-normal effects for rapid stabilization or agile responses to disturbances. Here, we extend this analysis to investigate how these non-normal dynamics influence the system's topological structure.

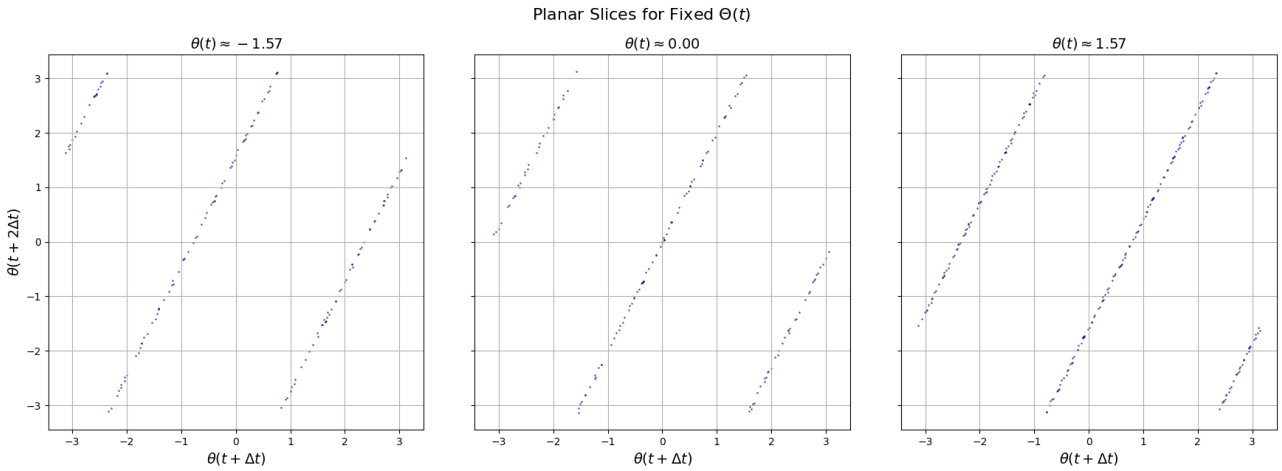


Figure 2: 2D projections of the 3D torus by fixing $\theta(t)$ at three different values. Each subplot shows $\theta(t + \Delta t)$ vs. $\theta(t + 2\Delta t)$for all velocities. Points form straight lines with slope approximately equal to 2, reflecting the time delay ratio. These lines demonstrate how varying speed influences the phase trajectory, while also illustrating that not all states in the 3-torus are accessible through velocity variation alone.

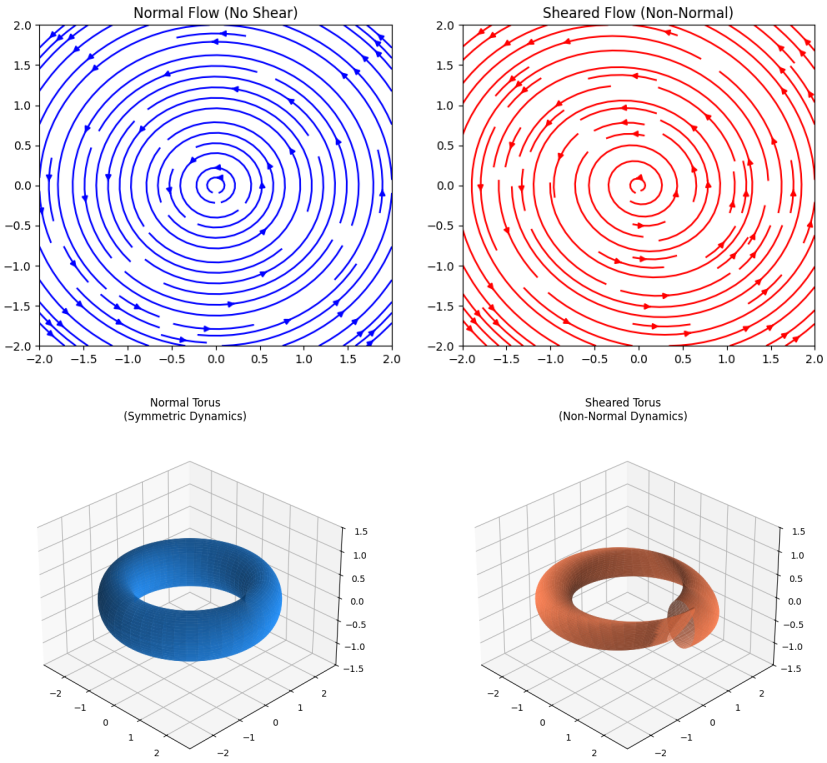


Figure 3: The presence of non-normal dynamics, visually demonstrated by the asymmetric flow patterns in the sheared torus, fundamentally alters the system's topological organization by introducing directionally biased interactions. In contrast to the symmetric dynamics of the normal torus (where perturbations decay uniformly), the non-normal system exhibits two key features: (1) transient amplification along specific phase-space directions, and (2) non-orthogonal eigenmodes that create preferred pathways for signal propagation.

A simple yet effective method to engineer non-normality in a dynamical system is through the introduction of an asymmetric coupling matrix

$$W = \begin{bmatrix} 0 & \pi\varepsilon \\ -\pi/\varepsilon & 0 \end{bmatrix}$$

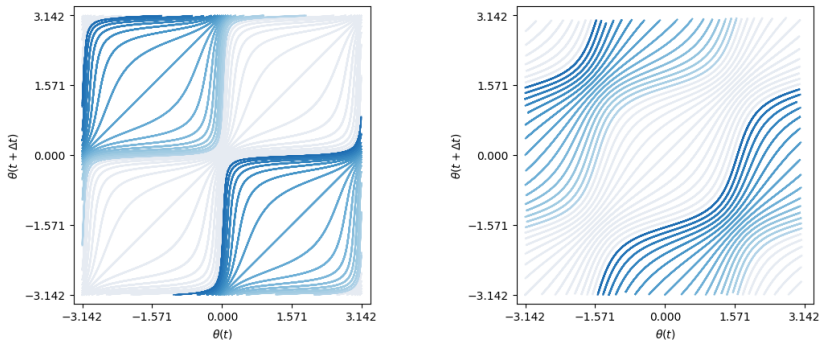


Figure 4: Demonstrates how the shear parameter ε controls non-normality. For $\varepsilon = 0.2$ (left), the system exhibits stronger asymmetry (higher Henrici index), manifesting as skewed trajectories in $\theta(t)$. At $\varepsilon = 1.8$ (right), the dynamics approach symmetry (lower Henrici index), consistent with the matrix structure W. This confirms that non-normal effects scale with the off-diagonal coupling ratio $1/\varepsilon$.

We further analyzed how symmetric and asymmetric coupling strengths govern system dynamics through

$$W_{\text{symmetric}} = \begin{bmatrix} 0 & \pi\varepsilon \cdot x \\ -\pi/\varepsilon \cdot x & 0 \end{bmatrix} \qquad W_{\text{asymmetric}} = \begin{bmatrix} 0 & \pi\varepsilon \cdot x \\ -\pi/\varepsilon \cdot y & 0 \end{bmatrix}$$

While our current analysis focused on minimal non-normal systems, we plan to extend this framework to coupled oscillator networks where asymmetric interactions model biologically plausible excitatory ($C > 0$) and inhibitory ($C < 0$) connections. The proposed block matrix structure introduces controlled directionality through distinct shear parameters $(\varepsilon_1, \varepsilon_2)$ and a cross-coupling term C, enabling study of how non-normality emerges in hierarchical neural circuits.

$$W = \begin{bmatrix} 0 & \pi\varepsilon_1 & 0 & 0 \\ -\pi/\varepsilon_1 & 0 & 0 & 0 \\ 0 & C & 0 & \pi\varepsilon_2 \\ 0 & 0 & -\pi/\varepsilon_2 & 0 \end{bmatrix}$$

3.3 Adding more legs: gait dynamics

In the more realistic problem of hexapod locomotion, we describe the state of the system using the vector $\mathbf{Y} = [x, L_1, L_2, L_3, R_1, R_2, R_3]^\top$, where all coordinates are defined with respect to the fly's centre of mass. The variable $Y_0 = x$ is a ghost degree of freedom that may represent, for instance, the position of leg L_1 perpendicular to the direction of locomotion. The remaining components Y_j for $j = 1, \ldots, 6$ denote the longitudinal position of the six legs along the direction of motion. The subscripts L_i and R_i refer to the left and right legs, respectively, ordered from front to back with $i = 1, 2, 3$. We consider two coordination patterns that generate distinct gaits [1]:

- Tripod: legs on opposite sides of the body move in synchrony in alternating triplets. Specifically, the front left (L_1), middle right (R_2), and hind left (L_3) legs swing together, while the complementary triplet (R_1, L_2, R_3) swings in anti-phase. This can be summarised as $L_1 = R_2 = L_3$ and $R_1 = L_2 = R_3 = -L_1$.
- Wave: each leg moves individually, with its motion coordinated by the leg that precedes it, forming a sequential wave-like pattern along the body axis.

Because the limb motions are coupled, one oscillator is sufficient to determine the full tripod pattern. Indeed, $\mathbf{Y}$ can be written in terms of a reduced variable $\mathbf{X} = (x, L_1)$ via the mapping

$$\mathbf{Y} = M\mathbf{X}\,, \quad M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}. \tag{2}$$

If $\mathbf{X}$ evolves under

$$\dot{\mathbf{X}} = W\mathbf{X}\,, \quad W = \begin{bmatrix} 0 & \pi \\ -\pi & 0 \end{bmatrix}, \tag{3}$$

then the full state obeys

$$\dot{\mathbf{Y}} = W_{\mathrm{t}}\mathbf{Y}\,, \quad W_{\mathrm{t}} = MWM^{-1}\,, \tag{4}$$

where M^{-1} is a left inverse of M. Suitable initial conditions are of the form $\mathbf{Y}_{0,\mathrm{t}} = [\sqrt{1-A_1^2}, -A_1, A_2, -A_3, A_1, -A_2, A_3]^\top$ with $A_i \in [0, 1]$. The simplest case, $A_1 = A_2 = A_3 = 1$, is not good for visualisation due to excessive curve overlap, so we choose the amplitudes randomly to obtain the time evolution under this coordination pattern (Fig. 5a). As expected, both the intra- and inter-pair state-space plots collapse to lines, reflecting perfect anti-phase coordination typical of the tripod gait.

In contrast, the wave gait is described by the linear dynamical system

$$\dot{\mathbf{Y}} = W_{\mathrm{w}}\mathbf{Y}\,, \quad W_{\mathrm{w}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\pi & 0 & 0 & 0 & \pi \\ 0 & \pi & 0 & -\pi & 0 & 0 & 0 \\ 0 & 0 & \pi & 0 & -\pi & 0 & 0 \\ 0 & 0 & 0 & \pi & 0 & -\pi & 0 \\ 0 & 0 & 0 & 0 & \pi & 0 & -\pi \\ 0 & -\pi & 0 & 0 & 0 & \pi & 0 \end{bmatrix}. \tag{5}$$

Here, non-zero off-diagonal elements implement the sequential activation and inhibition of limb movement: each leg j is excited by leg $j-1$ and suppressed by leg $j+1$, with indices wrapping cyclically. An appropriate initial condition for this regime is the "cascade" configuration $\mathbf{Y}_{0,\mathrm{w}} = [0, 1, 0.6, 0.2, -0.2, -0.6, -1]^\top$, which gives rise to a pattern where the limbs oscillate with phase delays (Fig. 5b). In this case, both the intra- and inter-pair state-space plots display elliptical trajectories, indicating stable phase-lagged oscillations characteristic of wave-like coordination.

Interestingly, *Drosophila* have been observed to switch from tripod to wave gaits as walking speed increases [1, 2]. To model this behaviour, we introduce a composite linear dynamical system that

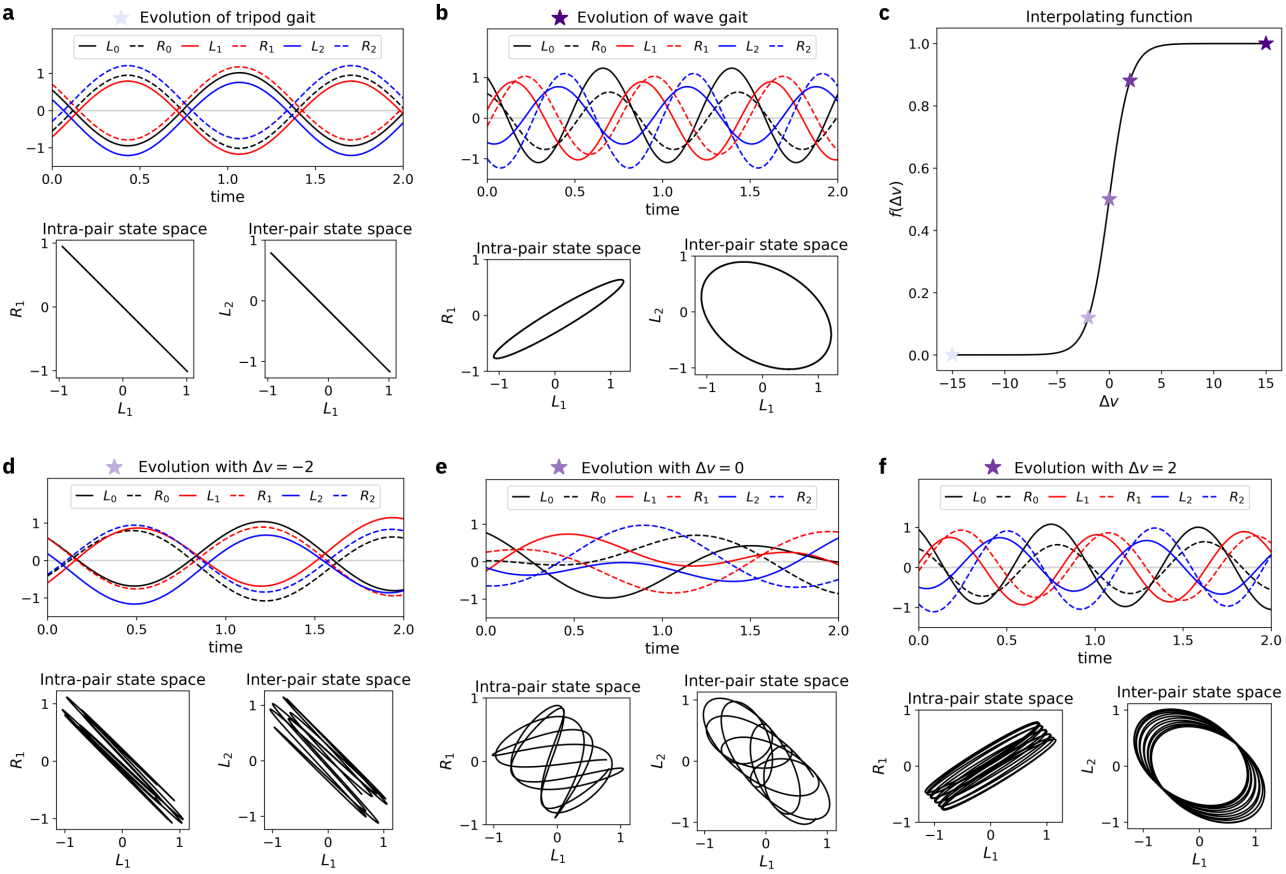


Figure 5: **Linear dynamical systems for hexapod coordination and their interpolation across walking speeds. a,** Time evolution and state-space projections for the tripod gait, generated by a low-dimensional linear system with strong inter-leg synchronisation. **b,** Wave gait dynamics arising from sequential activation and inhibition, producing a phase-delayed travelling-wave pattern across limbs. **c,** Sigmoid function $f(\Delta v)$ governing the smooth transition between tripod and wave gaits as a function of the speed difference $\Delta v = v - v_c$. Coloured stars highlight the location of the representative plots in the other panels. **d-f,** Gait evolution under the interpolated system for $\Delta v = -2$, 0, and 2, respectively. At these intermediate points, state-space trajectories broaden and lose regularity, reflecting partial desynchronisation and the emergence of intermediate coordination states.

interpolates between the two coordination regimes based on the difference $\Delta v = v - v_{\mathrm{c}}$ between the fly's speed v and the critical value v_{c}:

$$\dot{\mathbf{Y}} = W_{\mathrm{c}}(\Delta v)\mathbf{Y}\,, \quad W_{\mathrm{c}}(\Delta v) = [1 - f(\Delta v)]W_{\mathrm{t}} + f(\Delta v)W_{\mathrm{w}}\,. \tag{6}$$

Here, the function $f(\Delta v) = 1/\left(1 + e^{-\Delta v/\sigma}\right)$ smoothly weights the contribution of each gait-generating matrix, with the parameter σ controlling the sharpness of the transition (Fig. 5c; we take $\sigma = 1$).The corresponding initial condition is given by the same interpolation: $\mathbf{Y}_0 = [1 - f(\Delta v)]\mathbf{Y}_{0,\mathrm{t}} + f(\Delta v)\mathbf{Y}_{0,\mathrm{w}}$. As Δv increases, the system evolves continuously from tripod coordination ($\Delta v \ll 0$, Fig. 5a) to wave dynamics ($\Delta v \gg 0$, Fig. 5b), with an intermediate regime exhibiting mixed features (Figs. 5d-f). This transition is reflected in the structure of the state-space trajectories: while extreme values of Δv yield simple geometric shapes characteristic of the limiting gaits, intermediate values produce more complex, broadened orbits. These distortions reveal transient coordination and partial desynchronisation, illustrating how gradual changes in coupling can give rise to a continuum of gait dynamics.

3.4 Topological Analysis of Oscillatory Gait Dynamics on Torus Manifolds

Multi-legged locomotion involves complex coordination patterns between individual limbs that create characteristic gaits. Understanding these coordination mechanisms requires mathematical tools that can capture the underlying dynamical structure of oscillatory systems. This simulation investigates the topological properties of phase spaces generated by oscillatory gait models, comparing single-leg dynamics with coordinated multi-leg systems exhibiting tripod and wave gaits, as developed in 3.3. Using delay embedding techniques, we reconstruct phase space attractors and visualise them on torus manifolds to characterise the geometric and topological complexity of different locomotor patterns. The simulation implements coupled oscillator models where individual leg movements are represented as phase oscillators, and different gait patterns emerge from specific coupling architectures described previously. By analysing how these systems evolve across different velocities and visualising their trajectories on torus surfaces, we can quantify the topological complexity of each gait pattern and understand their fundamental mathematical properties.

The single-leg oscillatory system follows $\frac{dx}{dt} = vWx$, whilst the multi-leg systems for hexapod locomotion use the combined matrix $W_c(\Delta v)$ for tripod, wave, and intermediate gaits via $\frac{dY}{dt} = vW_c(\Delta v)Y$. Systems were integrated numerically (Dormand-Prince method) over $t \in [0, 10]$s, with velocity v varied (e.g. $v \in [1, 2]$). Time-delay embedding, $y(t) = [s(t), s(t + \tau)]$ with $\tau = \pi/(2v)$ (approximately one-quarter period), was applied to reconstruct the phase space from relevant time series. Trajectories were then visualised on the fundamental rectangle $[-\pi, \pi] \times [-\pi, \pi]$ and as paths on 3D toroidal manifolds.

Figure 6 illustrates phase space trajectories projected onto the fundamental rectangle. The single-leg system shows simple patterns, while the tripod gait displays characteristic figure-eight patterns reflecting anti-phase coordination between alternating leg groups. The wave gait exhibits more complex sequential activation patterns, indicative of a higher-dimensional structure. These trajectories, computed over a velocity range, demonstrate how phase space exploration varies with speed. Figure 7 presents 3D torus embeddings of these delay-embedded dynamics. The tripod gait creates trajectories wrapping around both major and minor torus circumferences, with density variations reflecting bimodal coordination. The wave gait shows more intricate winding patterns, exploring diverse topological regions corresponding to its sequential coordination mechanism. The chosen delay τ effectively unfolds these complex phase relationships. Figure 8 displays phase space coverage density maps. The tripod gait shows distinct bimodal clustering, corresponding to in-phase and anti-phase regions due to

its two-group coordination. The wave gait exhibits more distributed coverage, reflecting the sequential nature of leg activation and its higher-dimensional underlying dynamics. These maps, from numerous sample points (e.g. 1000 over $v \in [1, 2]$), quantify frequently visited phase space regions.

Analysis of complexity metrics (e.g. phase space coverage area, trajectory variance) generally indicates that the wave gait has larger coverage and higher variance than the tripod gait, suggesting more diverse phase relationships and higher intrinsic dimensionality. Both multi-leg systems are more complex than the single-leg system. Velocity-dependent analysis further reveals that the tripod gait maintains a relatively stable topological structure across velocities, consistent with its robust coordination. In contrast, the wave gait can display more significant structural changes with velocity, reflecting sensitivity to timing, which might be beneficial for nuanced control.

Our results demonstrate that different gait patterns possess distinct topological signatures, quantifiable through delay embedding. The tripod gait's bimodal structure suggests robust coordination suitable for fast locomotion, while the wave gait's higher complexity and velocity sensitivity may provide finer motor control, consistent with its use at slower speeds or for precise maneuvering. The delay embedding approach provides a powerful tool for analysing multi-oscillator temporal dynamics, offering intuitive geometric insight into complex phase relationships in coupled systems. While the linear oscillator model has limitations (e.g. no nonlinearities, uniform coupling), this framework bridges dynamical systems theory with gait analysis, offering new perspectives on the mathematical foundations of biological locomotion. Future work could incorporate nonlinear models, heterogeneous coupling, sensory feedback, and advanced topological invariants like persistent homology.

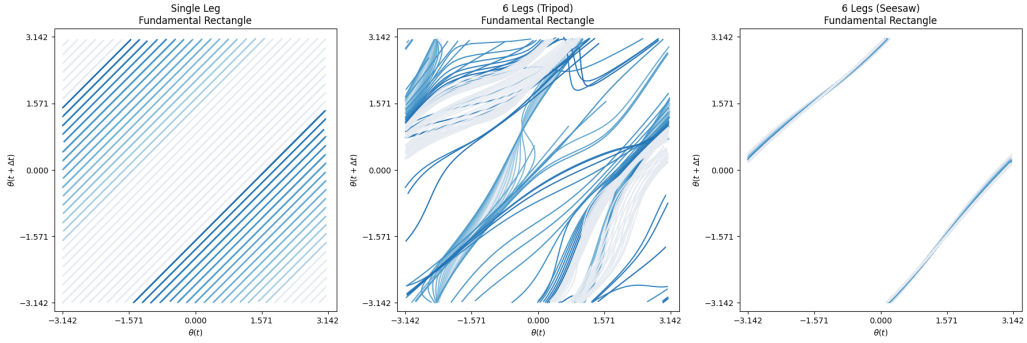


Figure 6: Phase space trajectories of different gait patterns in the fundamental rectangle $[-\pi, \pi] \times [-\pi, \pi]$. (Left) Single-leg oscillator showing simple circular trajectory. (Middle) Tripod gait displaying characteristic figure-eight patterns reflecting anti-phase coordination between alternating leg groups. (Right) Wave gait exhibiting complex sequential activation patterns with higher-dimensional topological structures. Trajectories computed over velocity range $v \in [1, 2]$ with colour indicating velocity magnitude.

4 Conclusions

This report investigated the topological underpinnings of walking, moving from a simple single-leg oscillator model to complex hexapod gait dynamics. We demonstrated that by employing idealised linear dynamical systems and time-delay embedding, the phase space of rhythmic leg movements can be understood as trajectories on toroidal manifolds of varying dimensions.

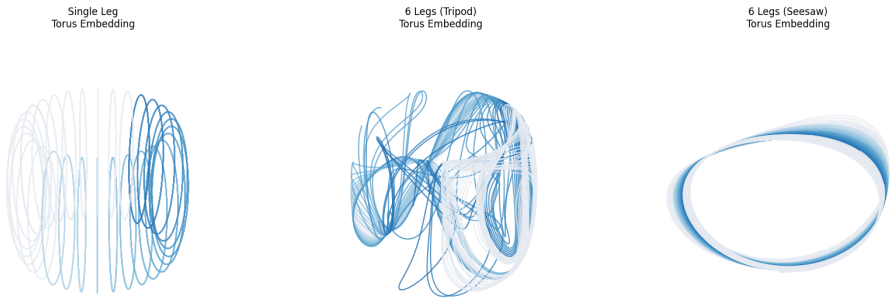


Figure 7: Three-dimensional torus manifold embeddings of delay-embedded gait dynamics. (Middle) Tripod gait trajectories wrapping around both major and minor torus circumferences with varying density patterns reflecting bimodal coordination structure. (Right) Wave gait showing intricate winding patterns and exploring distinct topological regions corresponding to sequential coordination mechanisms. Delay embedding applied with $\tau = \pi/(2v)$, revealing complex phase relationships in 3D space.

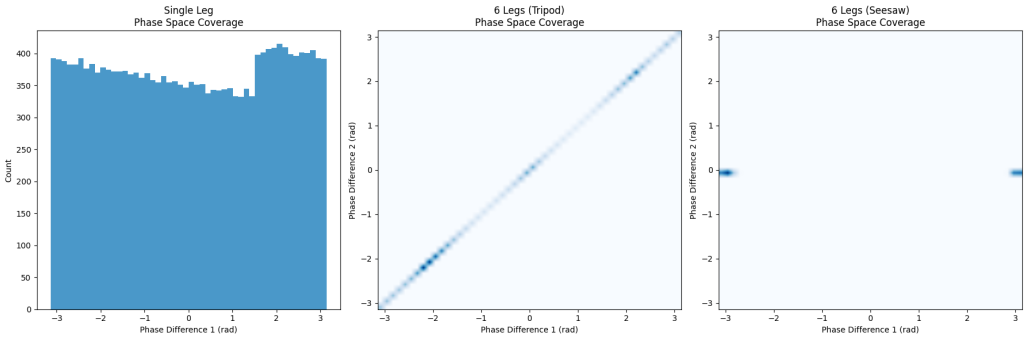


Figure 8: Phase space coverage density maps for different gait patterns. (Middle) Tripod gait showing distinct bimodal clustering with clear separation between in-phase and anti-phase coordination regions. (Right) Wave gait displaying more distributed coverage across phase difference space, reflecting the sequential nature of leg activation and higher-dimensional underlying dynamics. Heat maps represent trajectory density computed from 1000 sample points across velocity range $v \in [1, 2]$.

Our initial exploration showed how a basic 2D planar oscillator, representing a single leg's rhythmic motion (S^1), naturally extends to a torus ($T^2 = S^1 \times S^1$) with a single delay embedding. Further delays project the dynamics onto higher-dimensional tori (T^D), with variations in oscillation frequency (akin to walking speed) allowing the system to explore these toroidal surfaces more fully. We also illustrated how non-normal dynamics, introduced via asymmetric coupling (shear), can fundamentally alter the flow on these manifolds, providing a potential mechanism for rapid and agile responses in biological systems.

By extending our model to hexapod locomotion, we successfully captured distinct gaits—tripod and wave—using linear dynamical systems with different coupling matrices. A smooth interpolation function allowed us to model the transition between these gaits as a function of relative speed, revealing a continuum of coordination patterns characterised by gradual desynchronisation. The topological analysis of these gaits, through delay embedding and visualisation on torus manifolds, highlighted their unique signatures: the tripod gait exhibited a robust bimodal structure reflecting its alternating two-group coordination, while the wave gait displayed a more complex, higher-dimensional structure arising from its sequential leg activation.

These findings support our hypothesis that the latent structure of walking, particularly when analysed through delay embedding, is better characterised by toroidal geometries than by simpler cylindrical manifolds proposed in some prior work. This approach not only provides a framework for quantifying the complexity of different gaits but also sheds light on the functional implications of their topological properties, such as the robustness of tripod gaits versus the potential for finer control in wave gaits.

Future work could focus on incorporating non-linearities, more realistic neural and mechanical coupling, and sensory feedback into these models. This would allow for an investigation of stability under perturbations and the generation of more adaptive and robust locomotion dynamics. The presented methodology, combining idealised dynamical systems with topological data analysis, offers a promising avenue for uncovering the fundamental principles of locomotor control.

5 Acknowledgements

We would like to thank Angus Chadwick, Isabel Cornacchia and Arthur Pellegrino for their insightful lectures, as well as Max Eggl, Janko Petkovic and Laura Bernáez for organising the workshop.

References

[1] Brian D DeAngelis, Jacob A Zavatone-Veth, and Damon A Clark. The manifold structure of limb coordination in walking drosophila. *Elife*, 8:e46409, 2019.

[2] Anne Wosnitza, Till Bockemühl, Michael Dübbert, Henrike Scholz, and Ansgar Büschges. Inter-leg coordination in the control of walking speed in drosophila. *Journal of experimental biology*, 216(3):480–491, 2013.

How does the brain keep track of time? Duration manifold discovery via dynamical systems

Daniel Torres[*], Dario Sannino[†], Gina Schumacher[§], Niccolò Guglietta[¶]

[*]Instituto de Neurociencias de Alicante. [†]University of Bonn Medical Center.
[§]University of Cologne. [¶]University of Bologna.

Abstract

Flexible temporal control is a fundamental aspect of human cognition, enabling behaviors that are not directly tied to immediate sensory inputs or motor events. This capacity underlies complex functions such as speech, decision making, and planning. To better understand the neural mechanisms supporting flexible timing, we reanalyzed electrophysiological data from the medial frontal cortex (MFC) collected during a motor timing task, originally presented by Wang et al. (2018)[4]. While prior work suggested temporal scaling as a key mechanism, we explored whether more complex neural dynamics are at play. We applied Manifold Discovery via Dynamical Systems (MDDS), along with PCA, t-SNE, and temporal generalization analysis, to uncover low-dimensional structures and control strategies underlying population activity. Our results suggest that, although temporal scaling contributes to the observed patterns, flexible timing is supported by a richer interplay of dynamic trajectories and condition-specific control. These findings highlight the importance of low-dimensional dynamical models in uncovering how the brain flexibly regulates time.

1 Introduction

Time representation in the brain underlies critical brain functions such as anticipation, motor coordination, deliberation, and imagination. Flexible temporal control enables behaviours unrelated to immediate sensory or motor events and can unfold at different timescales. To support this flexibility, the brain must control the dynamics of ongoing patterns of neural activity.

The dominant theories of time representation such as clock-accumulator models, oscillation-based models, and population clock models, struggle to explain the complex, heterogeneous neural responses observed in certain regions of the brain, like the medial frontal cortex (MFC), where the neural activity does not conform to linear ramping or oscillatory patterns.

In the experiments by Wang et al. (2018)[4], rhesus monkeys performed a cue-set-go task to produce different time intervals. The researchers found that when the monkeys produced longer intervals, neural population activity in the MFC evolved along an invariant neural trajectory, but at a lower speed. Speed modulation of neural activity was also observed in the experiments of Egger et al. (2019)[1], where monkeys had to perform a 1-2-3-go task. The authors found that the speed modulation of the neural activity is inversely proportional to the delay. Further insights by Remington et al. (2018)[3] suggested that modulation of trajectory speed could be achieved by tuning initial conditions or through tonic inputs, like thalamic signals, that adjust the gain of the system.

These findings led to the proposal that flexible time is achieved via temporal scaling, or the control of the rate of change in the neural state. This view supports the idea that neural circuits function as

dynamical systems, where high-dimensional neural activity evolves according to underlying dynamical rules. Dimensionality reduction techniques have shown that these complex neural responses often lie on low-dimensional manifolds embedded within the high-dimensional state space.

Given the nature of the experimental data of Wang et al. (2018)[4], and the description of neural activity as a dynamical system, we sought to utilize Manifold Discovery via Dynamical Systems (MDDS), proposed in the paper of Pellegrino et al. (2025)[2]. This framework allows to directly infer the low-dimensional manifold and the underlying dynamical system that generates the recorded neural activity. This approach may help to clarify how complex, heterogeneous single-neuron responses collectively give rise to the observed population-level speed control and provide a rigorous, data-driven characterization of the underlying dynamical mechanism for flexible timing.

2 Methods

2.1 Experimental context

The data used in this project originate from a motor timing task described by Wang et al. (2018)[4], in which two adult rhesus monkeys were trained to reproduce specific time intervals using either a saccadic eye movement or a hand button press. Each trial began with the presentation of coloured fixation cues that simultaneously specified the required motor effector and the target interval (from 800 to 1500ms). After a variable delay, a Set cue indicated the start of the internally timed interval, which the animal reproduced via a movement (Fig. 1).

Wang et al. (2018)[4] obtained electrophysiological recordings from the medial frontal cortex (MFC), caudate nucleus, and thalamic regions using 24-channel laminar probes. In total, 1967 units (69% putative single units) were recorded, although for our analysis we used a curated subset provided by the workshop organizers.

2.2 Dataset used in the workshop

The dataset we used was a pre-processed and averaged subset of the full recordings. It consisted of neural activity from 54 neurons, all recorded from the medial frontal cortex. The recordings were organized into 10 conditions based on target durations ranging from 480 to 1200ms, in increments of 80 or 100ms: $T \in \{480,\ 560,\ 640,\ 720,\ 800,\ 800,\ 900,\ 1000,\ 1100,\ 1200\}\ ms^*$.

Each condition corresponded to trials with a specific target interval. All trials within a condition were averaged, producing a tensor of shape $condition \times time \times neuron$. This averaging reduced trial-level variability and emphasized the mean population dynamics associated with each interval.

In addition to the condition durations, we had access to a cue variable, which likely corresponds to the movement effector (eye or hand). While the original experiment involved both effectors at short and long durations (e.g. 800ms and 1500ms), the dataset appears to span shorter intervals (480-800ms) for eye movements and longer intervals (800-1200ms) for hand movements. However, due to the absence of detailed metadata, this mapping could not be confirmed with certainty. For most analyses, we focused on the eye movement subset (shorter durations), but we also explored the remaining conditions to

*800ms appears twice, likely from distinct cue subsets

assess potential differences across effectors. While this uncertainty does not affect the interpretation of the neural population dynamics per se, it limits the specificity of any effector-related conclusions.

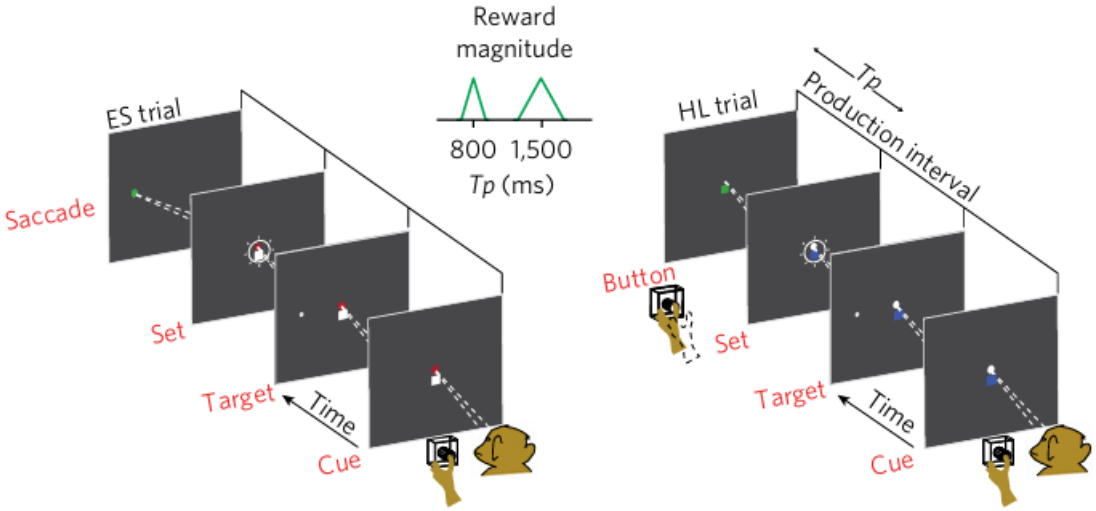


Figure 1: **Time production task from Wang et al. (2018)[4].** Monkeys were trained to reproduce time intervals using either a saccadic eye movement (left) or a hand button press (right), with visual cues indicating both the effector and the target duration. Each trial began with a fixation cue, followed by a Set signal marking the start of the internally timed interval. The two example trials shown — Eye + Short (800 ms) and Hand + Long (1500 ms) — are taken from the original paper. In our analysis, we used pre-processed data spanning a broader range of target durations between 480 ms and 1200 ms. Figure adapted from Wang et al. (2018)[4].

2.3 Preprocessing approaches

To investigate how population neural dynamics vary with interval duration, we applied two distinct preprocessing strategies to the condition-averaged data. These were designed to preserve either the real-time temporal structure or to enable direct comparison across conditions in a shared latent space:

Cut trials: In this approach, all neural responses were truncated to the duration of the shortest condition ($480ms$). This preserved the original real-time pacing and shape of neural dynamics, allowing for analyses grounded in actual temporal evolution. However, truncating longer trials inevitably discards late-time activity, which may contain important condition-specific or delayed features.

Warped trials: Here, each condition's activity was temporally rescaled ("warped") to span the same number of time bins, regardless of its actual duration. This normalization aligns the start and end of all trials and enables direct comparison of neural trajectories across conditions using dimensionality reduction techniques like PCA. However, the warping procedure transforms the temporal axis into a normalized "trial progression" scale, thereby distorting the real-time structure of the original recordings (slow dynamics in long trials are compressed).

These two preprocessing methods support the investigation of distinct hypotheses about how time is represented in neural population activity:

- Warped trials test whether the same latent trajectory is reused across different durations, with only the speed of traversal varying. This would suggest a temporal scaling mechanism, a hallmark

of dynamic systems encoding time by traversing a fixed manifold at different rates.

- Cut trials allow us to examine whether each duration recruits distinct neural dynamics, potentially reflecting condition-specific neural "paths" in latent space that are not simply scaled versions of one another.

By comparing both representations, we were able to explore whether the neural system encodes time through shared, temporally scaled dynamics, or through distinct trajectory structures tailored to each timing demand.

2.4 Manifold discovery via dynamical systems

To gain deeper insights into the dynamical system that generated the given, high-dimensional data we apply the method introduced by Pellegrino et al. (2025)[2]: Manifold Discovery via Dynamical Systems (MDDS). This approach claims that insights from the low-dimensional manifolds seen in experimental data can be transferred to the underlying geometry and dynamics of the system as the data naturally arise from dynamical systems principles. In detail it is argued that the time derivative of the trajectories of your time series data $\frac{\mathrm{d}x}{\mathrm{d}t} \coloneqq \dot{x}(t)$, representing the dynamics, near a point on the manifold lies in a vector field tangent to the manifold. Which implies that the manifold is defined by the tangent space together with a reference point. Thus, parameterizing dynamics on a manifold corresponds to expressing the system's evolution as a time-varying linear combination of vector fields that span the local tangent space. At each point in time, the system moves along a vector in the tangent space, determined by a set of coefficients that vary with time:

$$\dot{x} = c_1(t) f_1\left(x(t)\right) + \cdots + c_k(t) f_k\left(x(t)\right)$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system at time t (e.g. the activity of n neurons), each $f_i : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field, and k is the dimension of the tangent space — which also defines the intrinsic dimensionality of the manifold. The algorithm now minimizes the distance between the given data points and points along the trajectories of the model by finding the optimal parameters, meaning the c's and f's, and thereby infers the manifold.

3 Results

3.1 Condition-specific dynamics across preprocessing methods

To investigate how neural responses evolve over different interval durations, we visualized the firing rate trajectories of all 54 neurons across five conditions (480–800 ms) under two preprocessing strategies: cut and warped trials (Fig. 2). In the cut data, where all trials were truncated to 480 ms, the average population trajectories exhibit a consistent structure: an initial rise in firing rate followed by a peak and gradual decline. Notably, the timing of this peak shifts systematically across conditions, occurring earlier in shorter intervals and later in longer ones. This reflects a duration-dependent temporal profile in real time. However, because longer trials are truncated, some late-stage activity may be lost, potentially underestimating condition-specific divergence.

In contrast, the warped data, where all trials are temporally normalized to a common time axis, reveal a more coherent pattern: average firing rates across conditions become temporally aligned, and peak activity appears at similar normalized times. This alignment highlights a possible underlying structure shared across durations, consistent with the idea that neural activity evolves along similar trajectories in a normalized time frame but at different speeds in real time. The rightmost panel in each row reinforces this by superimposing the condition means, showing that warped trajectories are more similar in shape, suggesting a common dynamical template. These findings support the hypothesis that the brain may implement temporal flexibility not by recruiting entirely distinct dynamics for each interval, but by modulating the speed of a shared neural trajectory.

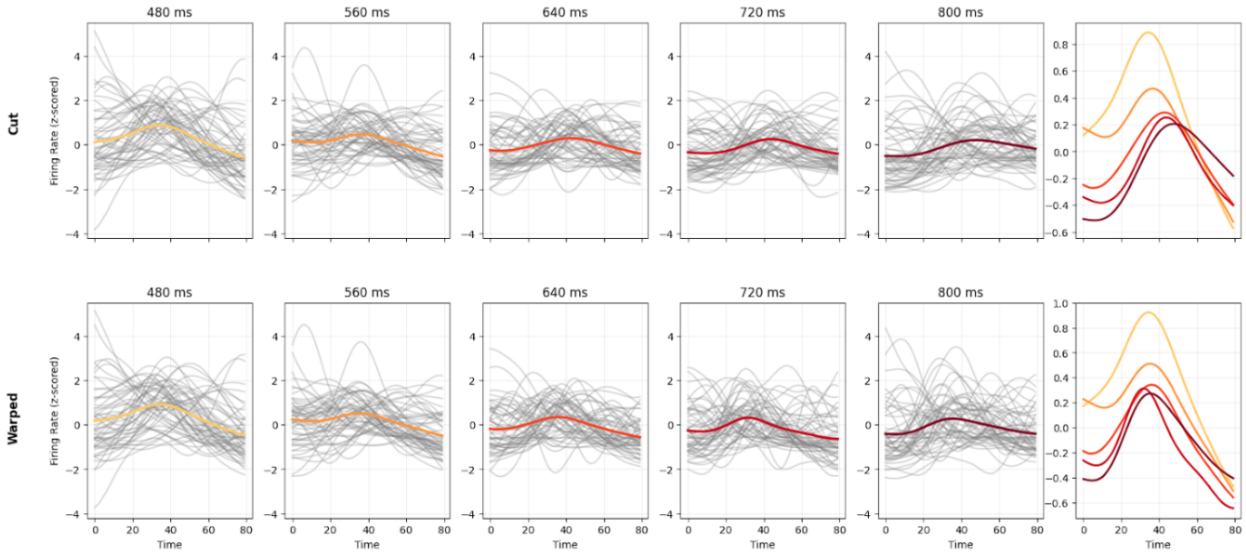


Figure 2: **Condition-specific neural dynamics under cut and warped preprocessing.** Each row shows firing rate traces for five target durations (480–800 ms) across all 54 neurons. Individual neuron responses are plotted in gray, with the condition-averaged trajectory overlaid in color (left to right: increasing duration). The top row corresponds to the cut dataset, in which all trials were truncated to 480 ms to preserve real-time pacing. The bottom row shows the warped dataset, where trials were temporally rescaled to match in length, aligning them across a normalized time axis. The rightmost panel in each row displays all condition means overlaid to highlight differences across durations. In both datasets, the average firing rate exhibits a peak whose timing shifts with the target duration, occurring earlier for shorter intervals. Warping aligns these peaks across conditions, revealing a shared temporal profile that is less apparent in cut data.

3.2 Insights from projections onto PCA spaces

Projection of the starting points $x(0)$ into two-dimensional PCA space shows that for the first five trials, the initial conditions can be separated along PC1 (Fig. 3, Sample 1 to 5). Same applies for the trials with longer time to memorize and hand movement (Fig. 3, Sample 6 to 10), whereas the two sets are separated along PC2.

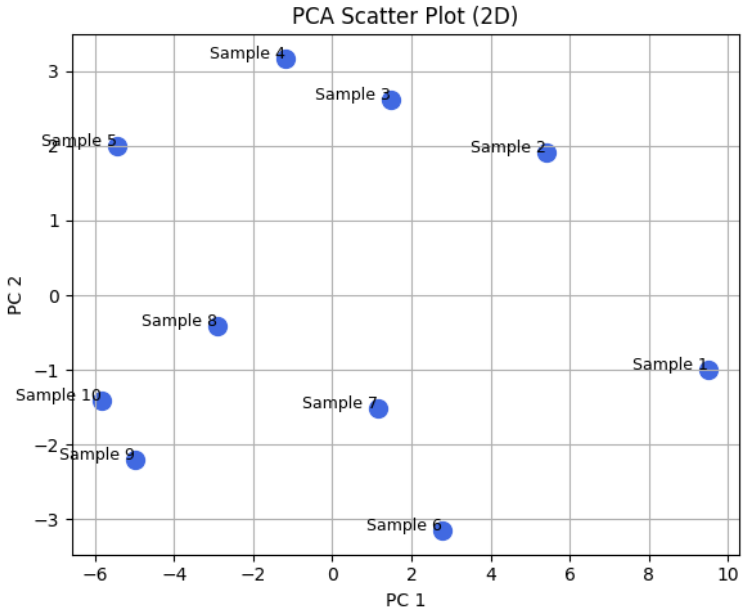


Figure 3: **Separation of initial conditions across timing conditions in PCA space.** Projection of the initial population activity $x(0)$ for all trials into a two-dimensional PCA space. The first five trials (shorter durations, eye movements) cluster along PC1, while the remaining five trials (longer durations, hand movements) are offset along PC2. This separation suggests that different timing conditions initialize the neural population in distinct regions of state space, potentially encoding information about the effector and target duration through initial conditions.

Plotting the whole trajectories into a three-dimensional PCA space indicates that the different conditions can be separated along the dimensions of PC1 and PC3 but hardly along PC2 (Fig. 4, Left). To understand the dynamics, we look at the full trajectories of all trials (e.g. the warped data) (Fig. 4, Right). The initial conditions lie further apart in 3D PCA space, but the trajectory evolution is similar for all trials and the end points lie much closer together, indicating convergence to a final state.

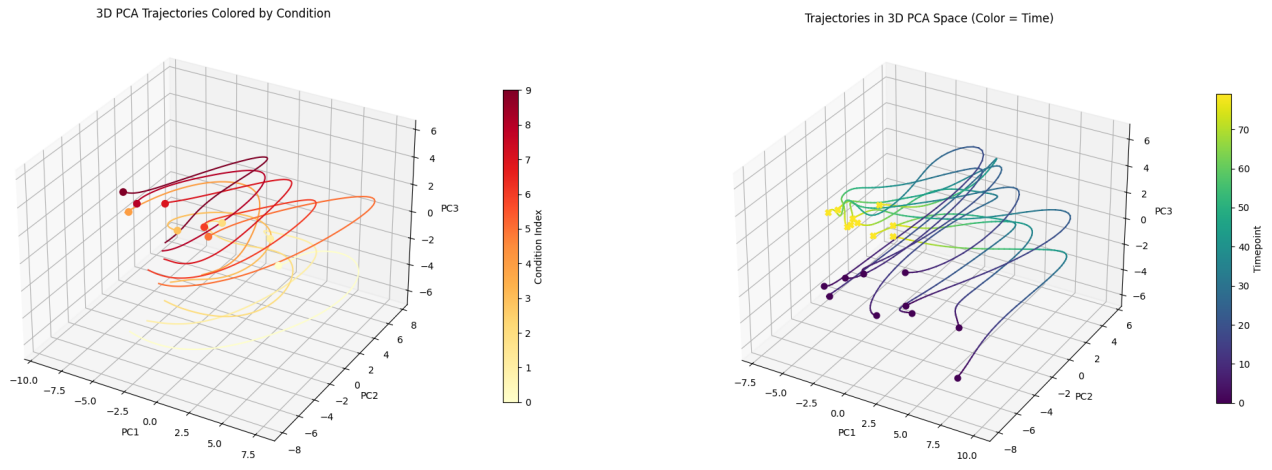


Figure 4: **Projection of trajectories for all trials into three-dimensional PCA space.** (Left) Trajectories of cut data, with different colors indicating different timing conditions. (Right) Trajectories of warped data with consistent color gradient representing the temporal progression of neural activity from trial start (dark blue) to endpoint (yellow).

To further explore the structure of these trajectories, we projected the time series data into a different low-dimensional embedding space: t-SNE. While PCA provides a linear dimensionality reduction that captures the directions of greatest variance, t-SNE employs a nonlinear approach optimized to preserve local structure.

Upon visual inspection, the trajectories in the t-SNE space exhibited clearer separation and were more easily distinguishable compared to those in the PCA space. Notably, the trajectories evolve similarly along t-SNE component 1, showing comparable trends across conditions. However, they are fully distinguishable along t-SNE component 2, which provides clear separation between different trajectory groups. This indicates that t-SNE better captures subtle, nonlinear relationships and clustering patterns in the data dynamics that PCA's linear projections may overlook. Consequently, t-SNE provides a more informative representation for differentiating distinct trajectory patterns within the time series data (Fig. 5).

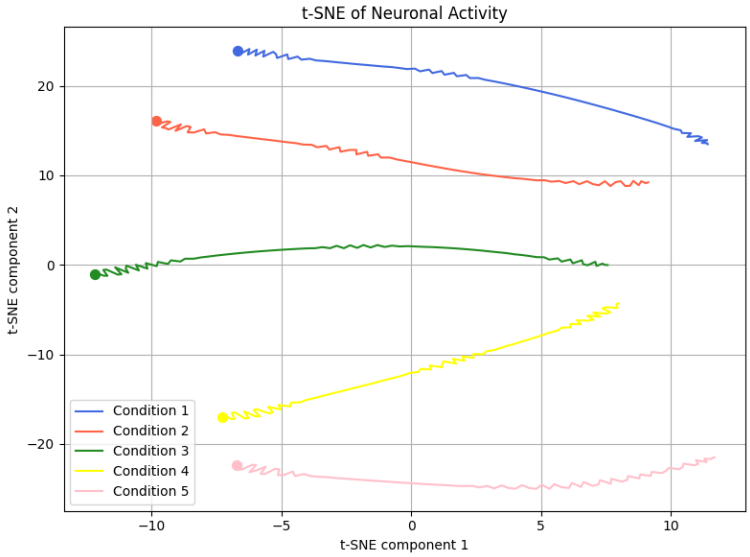


Figure 5: **Trajectories of neural population activity across five timing conditions projected into t-SNE space.** Trajectories originate from the left side and progress toward the right, reflecting the temporal evolution of each condition. Starting points are vertically aligned from top (condition 1) to bottom (condition 5), with each trajectory maintaining a distinct vertical position and exhibiting a similar overall shape. The clear separation along t-SNE Component 2 highlights the method's ability to distinguish between conditions and capture nonlinear structure in the neural dynamics more effectively than linear methods such as PCA.

3.3 Latent manifold structure in cut data

We applied MDDS to the neural activity obtained from cut trials, using an embedding dimensionality of 5 and a 2-dimensional control space. The resulting latent trajectories showed condition-dependent differences. Although all trials were temporally matched (480 ms), the neural dynamics for different conditions followed distinct paths through the latent space. This suggests that even within the same time window, the neural system recruits different dynamical strategies depending on the target interval (Fig. 6, latent trajectories and inferred manifold panels).

To evaluate the effectiveness of the learned dynamics, we compared models using linear and nonlinear vector fields (Fig. 6, top and bottom panels). Somewhat surprisingly, both approaches achieved

comparable levels of explained variance (approximately 0.77), indicating that each could reconstruct the neural activity with comparable accuracy. While explained variance reflects how well the model fits the data, it does not directly assess the quality or interpretability of the underlying dynamics. Nonetheless, the comparable values suggest that the additional flexibility of nonlinear dynamics may not be essential to capture the structure present in these recordings.

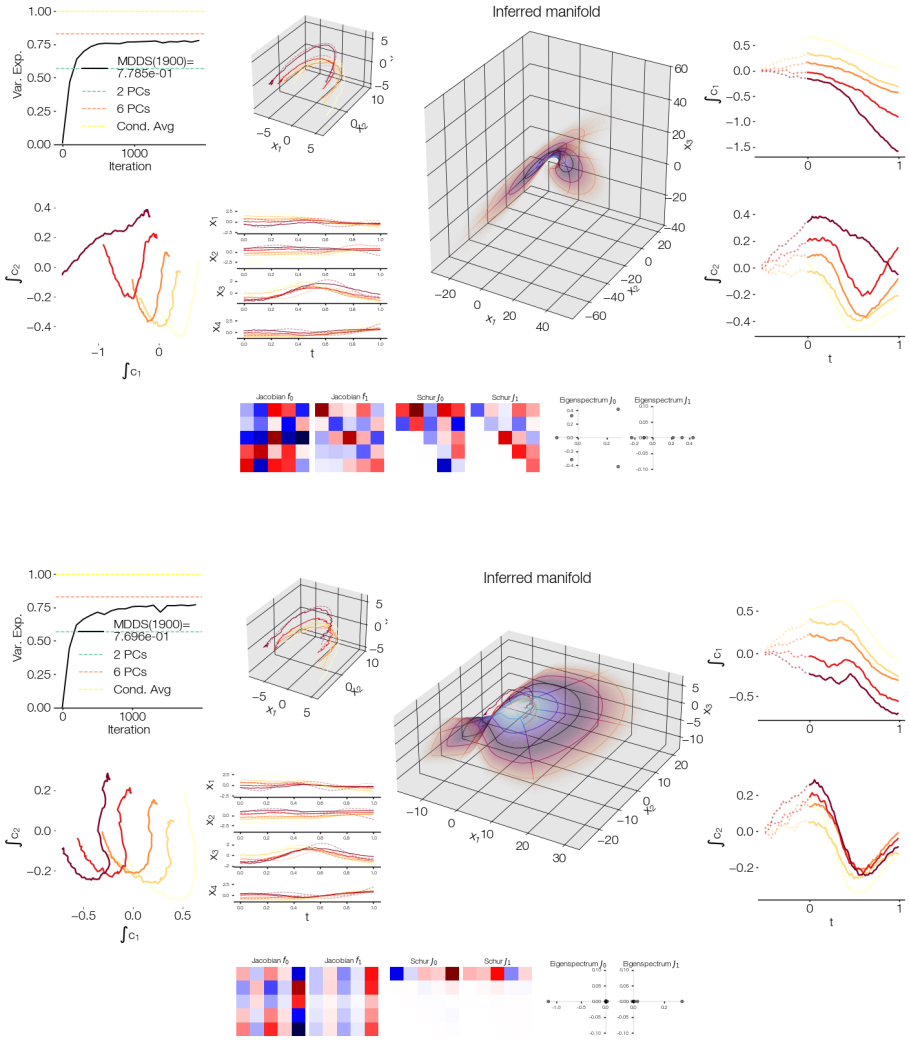


Figure 6: **Comparison of MDDS results using linear (top panel) and nonlinear (bottom panel) vector fields.** Each panel summarizes the output of the MDDS algorithm applied to neural activity from cut trials, using an embedding dimension of 5 and a control space of dimension 2. (Top panel): Results with linear vector fields. (Bottom panel): Corresponding results with nonlinear dynamics. (Leftmost plot): Explained variance as a function of training iteration. Dotted lines indicate variance explained by 2 principal components (PCs) and 6 PCs. Final MDDS performance is comparable (approximately 0.77) in both cases, suggesting that even linear vector fields provide an effective representation. (Phase plot): Trajectories in control integral space ($\int c_1(t)\,dt$ vs. $\int c_2(t)\,dt$), showing condition-specific control modulation. (Latent trajectories): 3D trajectories in the learned embedding space for each condition. PCA results are shown with dotted lines for comparison. (Inferred manifold): Smooth structure of the manifold with condition-colored trajectories, illustrating the geometry navigated by neural dynamics. (Control time courses): Cumulative integrals of the two control signals over time, per condition. (Reconstructed traces): Comparison between original and reconstructed neural activity for a subset of dimensions (x_1–x_4). (Jacobian and eigenspectra analysis): Linearization of the learned vector fields (f_0, f_1) provides insights into local stability and coupling.

To better understand how different strategies unfold over time, we examined the integrals of the control signals, $\int c_1(t)\,dt$ and $\int c_2(t)\,dt$, across conditions. These quantities represent the cumulative influence of each control component. However, the resulting patterns did not show a clear correspondence with condition duration — while some trajectories exhibited larger excursions in control space, there was no consistent trend (Fig. 6, Phase and control time courses plots).

This ambiguity suggests that differences across conditions may not simply reflect total control effort but instead involve more nuanced temporal profiles. To probe this further, we turned to an analysis of the control speed $\|c(t)\|$, which quantifies the instantaneous magnitude of the control signal and reveals how control evolves dynamically over the course of a trial.

Finally, to validate the learned dynamics, we reconstructed neural traces from the inferred latent trajectories and compared them to the original recordings. The close correspondence between model predictions and real data confirms that MDDS captures meaningful structure in the underlying neural activity (Fig. 6, Recontructed traces plot).

3.4 Control speed analysis from MDDS

To investigate how temporal structure is reflected in the inferred control signals, we analysed the magnitude of the control velocity $\|c(t)\|$, estimated from the latent trajectories recovered via MDDS. These control signals modulate how neural activity flows through latent space and are central to understanding how different durations are implemented by the system.

For the cut data, where all trials were truncated to the shortest interval ($480ms$), we approximated control signals using finite differences of the latent trajectories across time. After smoothing and computing the norm of these estimates, we obtained condition-specific velocity profiles. These profiles varied in both peak magnitude and temporal evolution, but did not show a clear monotonic relationship with condition duration (Fig. 7, left panel).

In theory, if flexible timing were achieved solely through temporal scaling, shorter durations would require higher average speeds through the same latent dynamics. However, our results revealed non-uniform peak times and inconsistent velocity magnitudes, suggesting that speed modulation alone cannot account for the observed behaviour.

In the warped data, trials were first time-normalized prior to MDDS fitting. To interpret the learned trajectories in real time, we reprojected the inferred latent factors by rescaling them according to each condition's original duration. This allowed us to reconstruct control velocity $\|c(t)\|$ as a function of real time.

The reconstructed profiles showed more systematic temporal structure: shorter conditions peaked earlier in real time, consistent with a translation-like temporal shift. The warping step reduced across-condition variability and aligned trajectories for easier comparison, though at the cost of smoothing over certain condition-specific dynamics present in the cut data (Fig. 7, right panel).

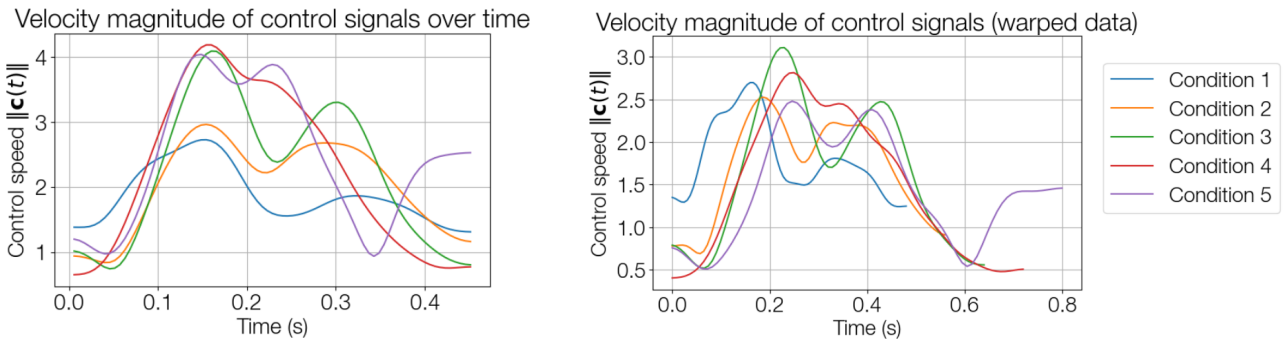


Figure 7: **Velocity magnitude of inferred control signals across time for each duration condition.** (Left) Control speed $\|c(t)\|$ computed from MDDS latent factors fit to cut data, where all trials were truncated to the shortest duration ($480ms$). Each curve represents a different timing condition. While peak magnitudes and dynamics vary across conditions, there is no clear monotonic relationship between duration and control speed. (Right) Control speed computed from warped data, in which trials were temporally normalized prior to MDDS fitting and then reprojected to real-time axes. Shorter-duration conditions exhibit earlier velocity peaks, suggesting a temporal shift rather than a uniform speed scaling.

3.5 Temporal generalization analysis

To evaluate how condition-specific information is encoded and evolves over time, we employed a temporal generalization analysis. This method tests whether a decoder trained on neural activity at a given time point can generalize to decode data from other time points. We used a linear support vector machine (SVM) to classify the interval condition based on neural population activity at each time.

We constructed a temporal generalization matrix in which the decoder is trained at each time point (rows) and tested at every other time point (columns). Each entry reflects classification accuracy, revealing how stable or dynamic the neural code is across time.

The generalization matrix obtained from the original cut data showed a strongly diagonal structure (Fig. 8, left panel). This pattern indicates that high decoding accuracy is achieved only when training and testing occur at similar time points, but generalization performance rapidly drops off for more distant times. Such time-specific encoding suggests that the population code for interval duration is highly dynamic, changing as the trial unfolds, rather than relying on static or sustained representations.

We repeated the same analysis using the data reconstructed from the latent factors inferred by MDDS. Although the overall structure of the generalization matrix was preserved, including a clear diagonal pattern, the decoding accuracy was noticeably reduced compared to the original data (Fig. 8, right panel). This decrease indicates that while MDDS captures essential temporal features of the data, the reconstruction may introduce some information loss, especially for fine-scale timing distinctions.

Nonetheless, the presence of a time-localized decoding profile in both the original and reconstructed data suggests that MDDS preserves key dynamical properties of the neural activity. The analysis confirms that the vector field inferred by MDDS does not collapse time or blur condition-specific evolution but instead retains temporally structured representations.

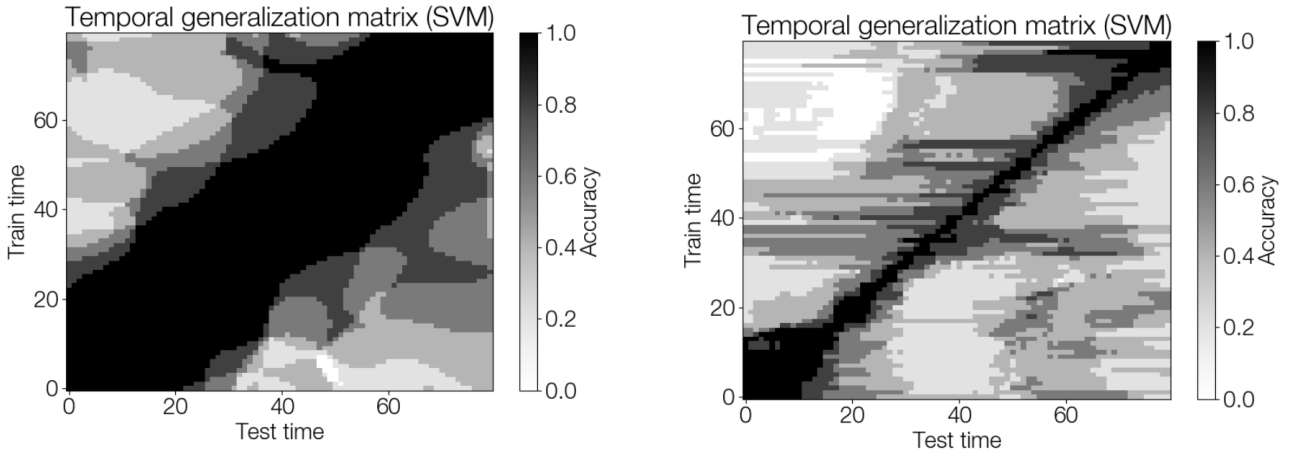


Figure 8: **Temporal generalization matrices for duration decoding.** (Left) Generalization matrix computed from the original cut neural data, showing accuracy of an SVM decoder trained and tested across all time points. Decoding accuracy peaks along the diagonal, indicating that duration information is encoded in a temporally local and dynamic manner. (Right) Generalization matrix computed from the MDDS-reconstructed data. While a diagonal structure is still evident, overall accuracy is lower and generalization to distant time points is reduced, suggesting partial loss of temporal specificity during latent trajectory reconstruction.

4 Conclusions

In this project, we investigated the neural mechanisms underlying flexible time production using electrophysiological recordings from the medial frontal cortex during a motor timing task. Motivated by the idea that time is encoded through structured neural dynamics, we applied Manifold Discovery via Dynamical Systems (MDDS) to uncover low-dimensional manifolds and the control mechanisms shaping population activity.

Initial analyses using PCA and t-SNE revealed that the neural activity corresponding to different timing conditions evolves along distinct yet similarly shaped trajectories. In the warped data, these trajectories notably converged to a common endpoint in PCA space, suggesting the existence of a shared final neural state despite different initial conditions and durations. This convergence may reflect a task-invariant goal state in the neural dynamics or a form of neural normalization that aligns behavioral outputs across conditions.

Beyond the convergence, we also observed a clear separation of trajectories across conditions, both in PCA and MDDS latent spaces. Each condition begins from a distinct initial point, indicating that the system initializes in a condition-specific manner before evolving toward a possible common attractor. This supports the idea that initial conditions may encode timing goals and serve as a critical parameter for modulating trajectory speed or shape. The geometry of the latent space thus appears to capture not only the shared structure of the timing task but also the condition-specific strategies used to achieve it.

Using MDDS, we modeled the dynamics of both cut and warped data, fitting vector fields and control signals on inferred manifolds. Linear and non-linear vector field models achieved similar levels of explained variance (approximately 0.77), but we note that high explained variance does not necessarily imply accurate recovery of the underlying dynamical structure. Our results therefore raise questions about how well variance-based metrics capture the interpretability or predictive quality of the inferred

dynamics, particularly when comparing linear and nonlinear models.

The inferred control signals revealed complex, condition-dependent profiles. However, no clear monotonic relationship with duration emerged, especially in the cut data. In warped data reprojected to real time, control speed $\|c(t)\|$ showed earlier peaks for shorter conditions, hinting at a temporal shift strategy rather than uniform speed scaling. These observations suggest that the brain may flexibly coordinate timing through more nuanced, condition-specific control regimes beyond simple modulation of trajectory speed.

Lastly, a temporal generalization analysis demonstrated that both the original and MDDS-reconstructed data encode interval information in a temporally local and dynamic fashion. While MDDS preserved the core temporal structure, a reduction in decoding accuracy suggests that some fine-grained information may be lost during manifold inference and reconstruction.

Overall, our results support the view that the medial frontal cortex realizes flexible timing through structured, low-dimensional neural dynamics. These dynamics likely involve condition-specific control patterns that converge toward shared goals while preserving trial-unique features. MDDS provides a valuable framework for uncovering these principles, though further work is needed to refine its interpretability and ensure its predictions reflect the underlying neural computations.

5 Acknowledgements

We would like to acknowledge the work of Wang et al. (2018)[4] as a foundational inspiration for this work. We are grateful to the members of the collaborating group and their institutions for their valuable input, as well as to the organizers of the event. Special thanks go to the members of the Angus Chadwick Lab for their insightful feedback and discussions. We also gratefully acknowledge the support of the Joachim Herz Foundation.

6 References

[1] Seth W. Egger et al. "Internal Models of Sensorimotor Integration Regulate Cortical Dynamics". In: *Nature Neuroscience* 22.11 (Nov. 2019), pp. 1871–1882. DOI: 10.1038/s41593-019-0500-6.

[2] Arthur Pellegrino et al. "Dynamical systems principles underlie the ubiquity of biological data manifolds". Unpublished manuscript. 2025.

[3] Evan D. Remington et al. "Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics". In: *Neuron* 98.5 (June 2018), 1005–1019.e5. DOI: 10.1016/j.neuron.2018.05.020.

[4] Jing Wang et al. "Flexible Timing by Temporal Scaling of Cortical Responses". In: *Nature Neuroscience* 21.1 (Jan. 2018), pp. 102–110. DOI: 10.1038/s41593-017-0028-6.

MDDS and Decoding on Center-out reach Task

Diletta Bartolini[1], Icaro Lorran Lopes Costa[2], Matteo Pizzinga[3], Mehdi Borjkhani[4]

Abstract

Understanding how neural population activity in motor cortex encodes movement parameters is fundamental to developing effective brain-computer interfaces (BCIs). We analyzed neural recordings from primary motor cortex (M1) of a monkey performing a center-out reach task to eight target directions using a joystick. Neural activity was examined using dimensionality reduction techniques including Principal Component Analysis (PCA), Uniform Manifold Approximation and Projection (UMAP), and Manifold Discovery via Dynamical Systems (MDDS) to uncover low-dimensional structure. We evaluated the effectiveness of these representations for target classification using Linear Discriminant Analysis and kinematic decoding via linear regression. MDDS revealed a cone-like neural manifold structure and significantly outperformed other methods, achieving Matthews correlation coefficients of $\rho_M \approx 0.87$ for target classification compared to $\rho_M \approx 0.47$ for raw neural data. Similarly, MDDS-generated representations improved kinematic decoding with linear regression performance ($R^2 \approx 0.66$ vs $R^2 \approx 0.57$ for raw data). A simple Brain-Computer Interface model demonstrated the potential utility of these manifold-based approaches, though with limited accuracy for trajectory prediction. These findings suggest that motor cortical activity evolves on curved, low-dimensional manifolds rather than linear subspaces.

[1]University of Pavia, FernUni Schweiz
[2]University of Bonn
[3]The Biorobotics Institute and Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, Pisa, Italy
[4]International Centre for Translational Eye Research (ICTER), Institute of Physical Chemistry, Polish Academy of Science, Warsaw, Poland

1 Introduction

Neuroscience studies how the brain processes information to produce behavior, and studying how variables of the world are represented in neural activity is central to tackling this question. A topic of particular interest is the study of how the brain generates movement. This question is often explored by recording monkeys performing movements and observing their neural activity changes as a function of these movements.

In literature many studies highlight that the dynamics on the manifold offer a framework to interpret motor control that reflects the low-dimensional, nonlinear, and adaptive nature of neural motor planning.[4] [6] [3] The Neural manifold emerges from the connectivity of neurons that impose constraint on the neural activity that the network can generate and on what can be learned[13] [10]. For extracting this manifold many linear[7] or non-linear[5] dimensionality reduction techniques were employed to detect patterns of co-activation of neurons (usually called neural "modes", for a review on dimensionality reduction see [4]).

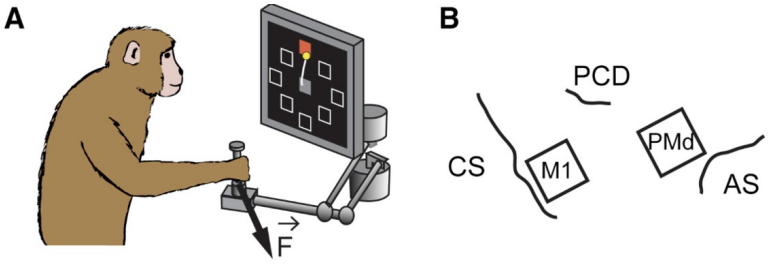


Figure 1: **A.** A monkey was trained to move a joystick with their hand to reach a target on the screen. **B.** In the meantime, populations of neurons were recorded in brain areas associated to movement.

In this study, we analyzed neural recordings from the primary motor cortex (M1) of a monkey performing a movement task(Figure 1). The task involved moving a cursor to one of eight target positions using a joystick. We focused on neural activity around movement onset ($t \in [-0.4, 0.8]s$), first examining the tuning of individual neurons to different reach directions.

We applied various dimensionality reduction techniques to uncover the underlying geometry of neural activity, including PCA and the recently developed MDDS. Our results revealed a curved, cone-like neural manifold with lower dimensionality than the number of recorded neurons. Leveraging these latent representations, we further explored their potential use for decoding or classify movements, testing their applicability for Brain-Computer Interface (BCI) applications.

Eventually we trained a Linear Dynamical Systems(LDS) to model the neural activity, evaluating its effectiveness as a surrogate for generating neural trajectories that are meaningful and retain information used for decoding the cursor's position.

Brain-Computer Interface

A Brain-Computer Interface (BCI) is an emerging technology that enables direct communication between the brain and external devices by translating neural data into machine-interpretable instructions. BCIs come in various forms, including invasive systems like Neuralink, which are surgically implanted on the brain's surface to allow direct control of machines through thoughts. However, non-invasive alternatives using EEG signals offer a less risky option for applications such as gaming and rehabil-

itation. In the context of this study, the BCI model should be able to control a joystick's direction based on visual targets displayed on a screen.

2 Methods

2.1 Neural Data

In this study we analyzed neural recordings during a movement task from a published work [11]. The subject – a monkey – is asked to move a cursor on the screen to reach one of eight target by moving a joystick. While the monkey performed the task, neural activity from 95 neurons was recorded from the primary motor cortex (M1).

2.2 Preprocessing

Neural activity was preprocessed to isolate task-related dynamics. Each trial generated a matrix $\mathbf{X} \in \mathbb{R}^{T \times N}$, where T denotes time points and N neurons, resulting in a 3D tensor $\mathcal{X} \in \mathbb{R}^{K \times T \times N}$ across K trials. Trials were grouped by target direction (8 conditions) and averaged per condition.

Spike trains and kinematic data were obtained from a dataset previously used in [11]. We focused on primary motor cortex (M1) activity. Spike trains were concatenated across channels and a zero-mean Gaussian filter ($\sigma = 4$ ms) was applied along the time axis to attenuate high-frequency noise.
Neural and kinematic signals were cropped to $[-0.4, 0.4]$ s relative to stimulus onset. Only trials belonging to the baseline (BL) epoch were retained. Neural data were baseline-corrected by subtracting the mean activity in the pre-stimulus window $[-0.4, 0)$ s and rescaled by the global standard deviation (with ϵ added for numerical stability).

2.3 Principal Component Analysis (PCA)

Following data preprocessing, we employed Principal Component Analysis (PCA) to characterize the neural population dynamics. PCA identifies orthogonal directions of maximal variance in high-dimensional data, enabling dimensionality reduction while preserving the most biologically relevant signals.

2.4 Uniform Manifold Approximation and Projection (UMAP)

Since the separation observed in PCA projections was not clearly distinguishable, we employed an alternative dimensionality reduction technique: Uniform Manifold Approximation and Projection (UMAP). UMAP is a powerful non-linear method particularly well-suited for visualizing high-dimensional data while preserving both local and global structures. Unlike PCA, which focuses on preserving linear relationships, UMAP maintains the underlying manifold geometry, making it especially effective for analyzing complex neural data.
The neural data originally resided in a high-dimensional space, with axes corresponding to neurons and time points. Reducing this dimensionality was essential to facilitate visualization and reveal potential patterns or clusters that are not apparent in the raw space. UMAP was selected for its ability to provide low-dimensional embeddings while preserving meaningful structure.

Once configured, UMAP was applied to the full neural dataset using the fit_transform method. This produced a two-dimensional embedding in which each point represents the high-dimensional neural activity at a specific time point, now projected into a 2D space.

To further examine condition-specific structure, we projected individual subsets, specifically conditions 0 and 1, using the pre-trained UMAP model. This ensured that the learned manifold structure was preserved across these subsets, enabling consistent comparisons.

Although Uniform Manifold Approximation and Projection (UMAP) provides a useful low-dimensional visualization of high-dimensional neural data, it fails to preserve the temporal continuity inherent in neural dynamics. Capturing these temporal aspects is critical for understanding the evolution of neural states during motor tasks. To overcome this limitation, we employed a more advanced approach—*Manifold Discovery via Dynamical Systems*, which integrates manifold learning with dynamical system modeling to capture both spatial and temporal patterns in the data.

2.5 UMAP Hyperparameter Search

To optimize the performance of UMAP for dimensionality reduction, a hyperparameter search was conducted using Optuna. This process is essential as UMAP's effectiveness significantly depends on several key parameters that influence the quality of the resulting embeddings.

The hyperparameters tuned include:

- `n_neighbors`: The number of neighbors considered in local manifold approximation, ranging from 5 to 50.
- `min_dist`: The minimum distance between points in the low-dimensional space, from 0 to 0.5.
- `n_components`: The dimensionality of the output space, explored between 2 and 10.

To ensure robust evaluation, a 5-fold cross-validation approach was employed. For each fold, UMAP was applied to transform the training data, followed by a validation step where a Linear Regression model predicted outcomes from the transformed features. This setup allowed for an unbiased assessment of the embedding quality across different splits.

The performance metric used for the optimization was the R2 score, which measures the proportion of variance explained by the model. The mean R2 score across all folds determined the best configuration, ensuring that the selected hyperparameters consistently performed well across different subsets of data.

2.6 Manifold Discovery via Dynamical Systems (MDDS)

To explicitly model temporal evolution, we applied Manifold Discovery via Dynamical Systems (MDDS). This approach merges manifold learning with dynamical system modeling, allowing us to recover both the low-dimensional manifold structure and the underlying temporal trajectories of neural activity. The central assumption is that neural responses evolve along smooth, low-dimensional manifolds governed by latent dynamics that can be described by differential equations. These parameters control various aspects of the model training, including the manifold and embedding dimensions, the penalty for regularization, the batch size, and the learning rate, among others. Specifically:

- Manifold dimension: 2 (to evaluate performance in a theoretically optimal low-dimensional space)
- Embedding dimension: 12 (to allow flexibility in capturing higher-order dynamics)

The model was trained using the `mdds.fit()` function, which optimizes the manifold representation by minimizing the reconstruction error and ensuring that the dynamics of the system are appropriately captured. The training process included regularization through the Frobenius penalty, which discourages overfitting by penalizing large deviations from the expected structure of the dynamics.
An important aspect of this analysis is to consider why a 2-dimensional manifold might be a theoretically sound choice for representing the data. From a theoretical standpoint, a 2D manifold is often sufficient for capturing the most significant features of high-dimensional data, particularly when the data lies on a lower-dimensional curved space. In the context of neural data, a manifold of higher dimensionality might introduce overfitting, potentially capturing noise rather than meaningful structure. Conversely, a 2-dimensional manifold may allow for the most relevant variability in the data to be captured, while excluding irrelevant fluctuations.

2.7 Classification of reached target

A Linear classification (Linear Discriminant Analysis - LDA) of the target angle was performed using data at the final time point, when the target is reached. The classifier was tested following a stratified 5-fold cross-validation scheme. The Matthews correlation coefficient[2]($\rho_M \in [-1, 1]$) was used as metric since it is more suitable for unbalanced datasets, as ours. Different neural data representations (raw, PCA, UMAP, MDDS) were used to evaluate which representation makes the information about the target angle linearly accessible.

2.8 Linear regression of cursor position

The dataset was composed also by the (x, y) position of the joystick used by the monkey to move the cursor on the screen (Figures 1 and 6a). We trained a linear regression model with least square to infer the original kinematic data for each time point (T samples, stored in a matrix $Y \in \mathbb{R}^{T\times 2}$) given the actual neural data representation (either raw, PCA, UMAP or MDDS).

2.9 Brain-Computer Interface Model

A brain computer interface model that would be applicable to this task is one where it can predict the next joystick position x_{t+1} given the previous position x_t. A minimal way in which this could be accomplished is with a linear model of the form

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

One way to train this model is to use the dynamics of the neural data $\{N_t\}$ as a proxy through the use of a decoder that can map neural data into position, as illustrated in figure 2. We have used a feed-forward model to accomplish this mapping.

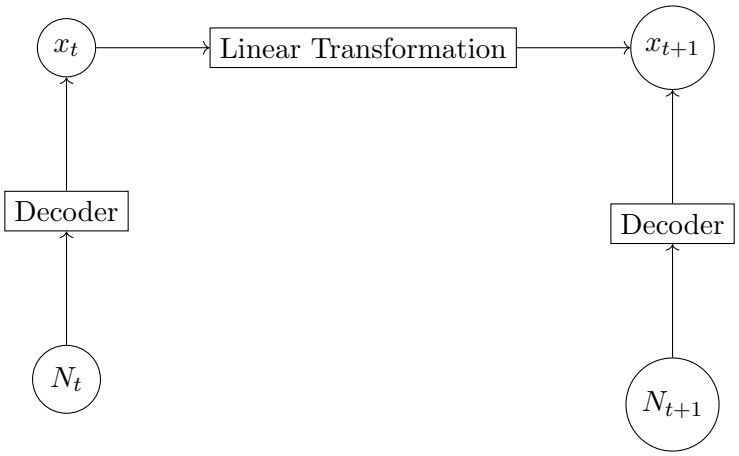


Figure 2: Schematic for the training of the brain-computer interface model using the neural data for proxy learning.

A minimal Brain-Computer Interface (BCI) model that can predict the next joystick position x_{t+1} based on the previous position x_t can be achieved through a linear transformation of the form

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

Where A represents the transformation matrix. To train this model, we leverage the dynamics of neural data $\{N_t\}$ as a proxy. This is achieved through the use of a decoder that maps neural activity to joystick position, as illustrated in Figure 2. Specifically, we employ a feed-forward neural network to establish this mapping. The transformation matrix A is subsequently learned via linear regression.

3 Results

3.1 Single unit's tuning curve analysis

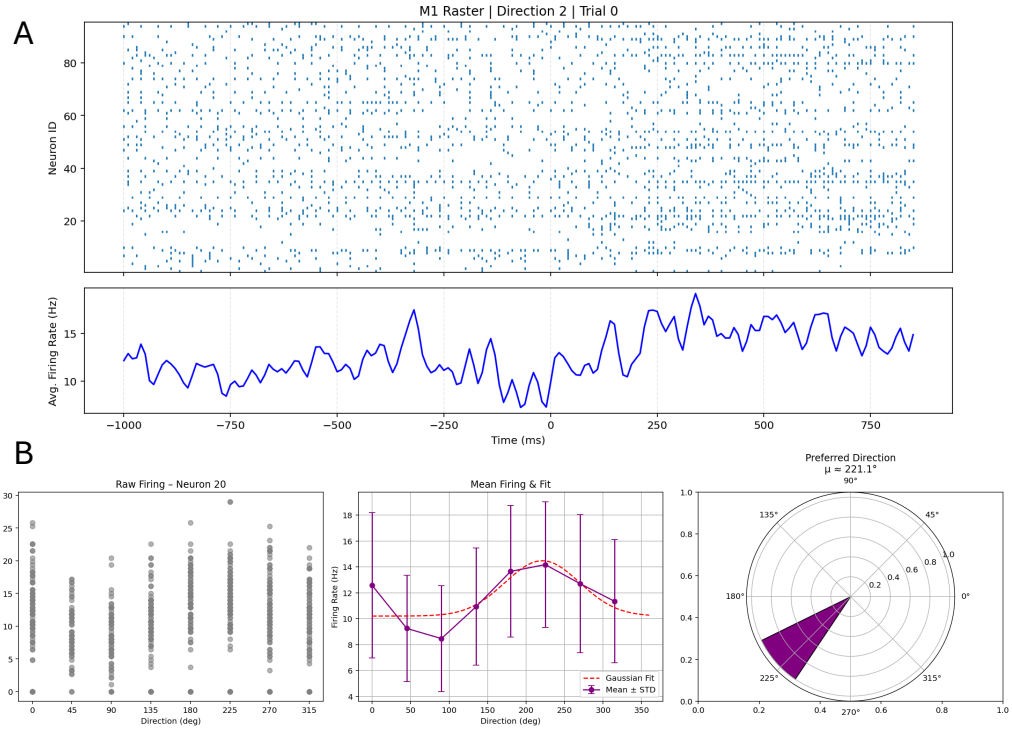


Figure 3: **Raster Plot and Population Firing Rate (Direction 2, Trial 0) and Direction Tuning for a Neuron. (A)** Top: Raster plot of 96 neurons. Bottom: Population average firing rate (bin-smoothed, Hz) across the trial. **(B)** Direction Tuning for Neuron 20. Raw firing rate across directions. Directional tuning curve with Gaussian fit. Polar plot of preferred direction.

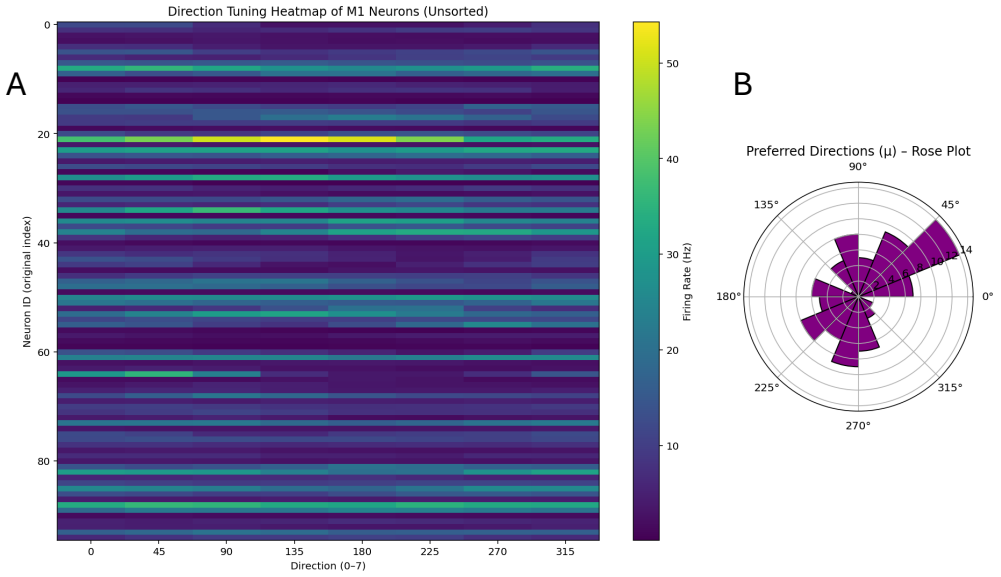


Figure 4: **Direction Tuning Heatmap and Distribution of Preferred Directions (Rose Plot).** Color scale = Firing rate (Hz). Each row = 1 neuron. Rose plot of tuning preference angles binned over 30° intervals.

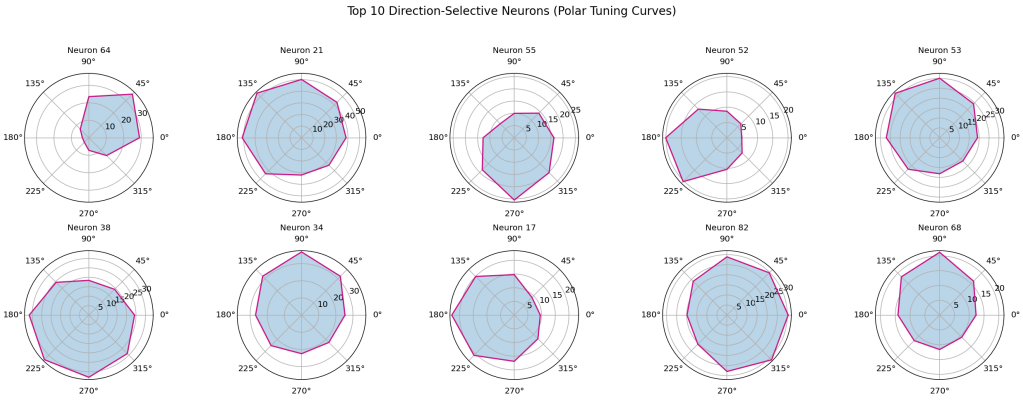


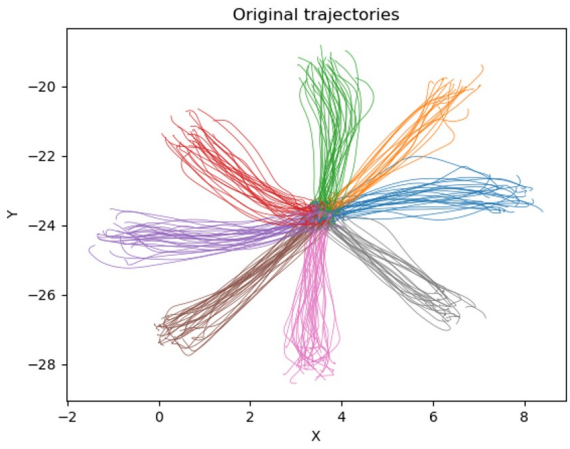
Figure 5: **Top 10 Direction-Selective Neurons (Polar Tuning Curves).** Each subplot: One neuron's polar tuning curve (max = peak response in Hz).

In Figure 3 shows the spiking activity of 96 M1 neurons during a single trial (Trial 0) in Direction 2 (target direction). The top panel presents a raster plot, where each row corresponds to a neuron and each dot marks a spike time. The bottom panel shows the average firing rate of the population over time, indicating modulations in activity around movement onset (time = 0 ms). Furthermore, Panel A in Figure 3 shows the raw firing rates of Neuron 20 across all directions and trials. Panel B displays the mean ± standard deviation of firing rates for each direction, with a Gaussian fit overlaid in red. Panel C is a polar representation indicating the preferred direction ($\mu \approx 221.1°$), suggesting a tuning toward that direction. In Figure 4, Firing rate responses of all M1 neurons (rows) are shown as a heatmap across 8 movement directions (columns). Each row corresponds to a neuron ID, and the color intensity represents the firing rate (Hz). Strongly tuned neurons display peak responses in specific

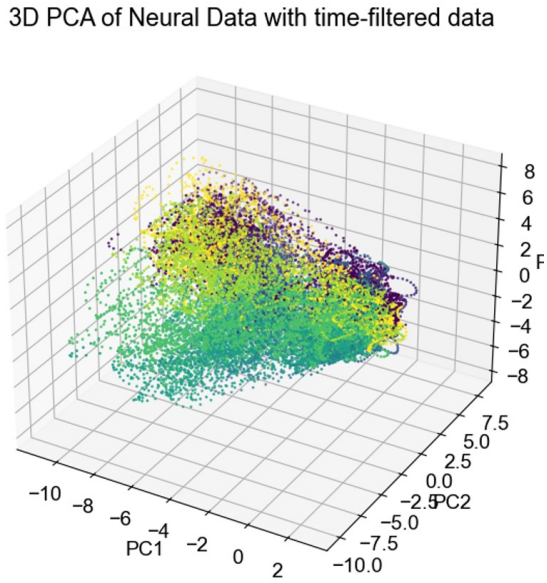
direction columns. Polar tuning curves for the top 10 direction-selective neurons. The blue-shaded area shows firing rate magnitude across 8 directions. These neurons exhibit clear peaks, each tuned to a specific preferred angle (Figure 5). A rose plot summarizing the distribution of preferred directions across the neural population. Each bar represents the number of neurons tuned to a specific angular bin, with a notable peak around 45° (Figure 4).

3.2 Dimensionality reductions techniques on Neural data

To assess whether distinct experimental conditions elicited differentiable motor patterns, we began with a qualitative examination of the kinematic data. We analyzed hand movement trajectories in the two-dimensional workspace, with each trajectory color-coded according to its corresponding experimental condition. These trajectories were extracted from a precisely defined time window aligned to the baseline (BL) epoch, ensuring a consistent comparison across trials while focusing on unperturbed movement execution. Figure 6a illustrates the resulting hand paths for all trials, with distinct colors representing the eight experimental conditions.



(a) Hand movement trajectories in the 2D workspace. Each trace represents the path followed during a single trial (n = 112 trials), colored by experimental condition (8 distinct targets). Trajectories were extracted during the baseline epoch (BL) and normalized to common start/end points.



(b) PCA applied to the full dataset, including all time points and conditions. The 3D plot shows the trajectories of neural activity over time, with each trajectory corresponding to a different condition.

Figure 6: Kinematic and neural trajectories during the center-out reach task.

We applied time-resolved PCA to investigate the temporal structure of neural dynamics throughout each trial. This analysis revealed condition-dependent trajectories in neural state space, supporting the idea of evolving, structured dynamics underlying the task (Figure 6b).

Although PCA revealed condition-specific dynamics, its linear nature limits its ability to uncover complex neural geometry. To overcome this, we applied UMAP, a nonlinear dimensionality reduction technique that preserves local neighborhood relationships and is more sensitive to intrinsic low-dimensional structure.

To explicitly capture the structure of neural trajectories over time, we employed the MDDS (Manifold-based Dynamical System Synthesis) method. MDDS infers low-dimensional latent dynamics from high-dimensional time series data, allowing us to model the temporal evolution of neural population activity. The results are summarized in Figure 7.

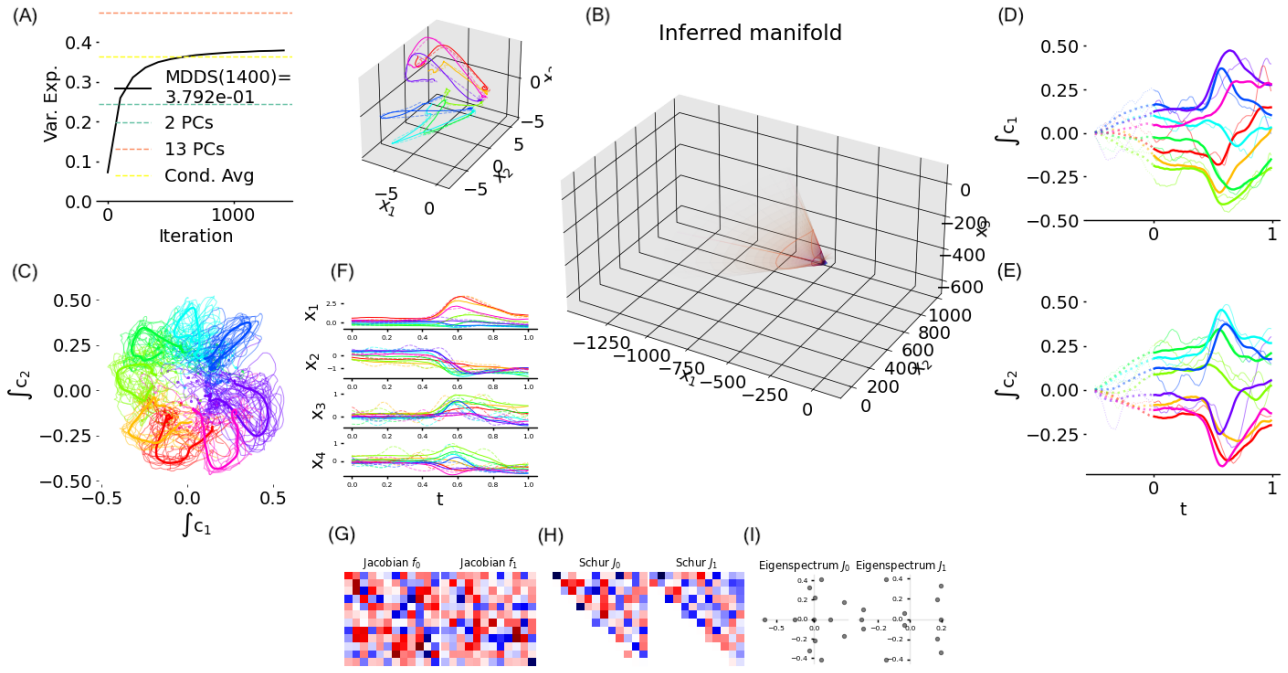


Figure 7: Dynamical manifold inference using MDDS. (A) Explained variance during training. (B) 3D trajectory plot and inferred latent trajectories. (C) Projections in the space of integrated control signals $\int c_1$ and $\int c_2$, colored by condition. (D,E) Time courses of integrated controls. (F) Reconstructed observed signals for selected dimensions. (G,H,I) Jacobians, Schur decompositions, and eigenvalue spectra of the learned dynamics.

In order, panel (A) shows the amount of explained variance across optimization iterations. While the first two principal components (PCs) account for only a limited portion of the variance, the 2D latent space inferred by MDDS explains approximately 38% of the total variance, outperforming linear PCA of the same dimensionality. This demonstrates the benefit of using nonlinear dynamic models for dimensionality reduction.

In Panel (B), the low-dimensional latent dynamics inferred by MDDS for each condition reveal structured and condition-specific trajectories in the learned latent space. The manifold exhibits a cone-like geometry, suggesting that the dynamics expand from a narrow initial region into condition-specific trajectories, following a smooth and continuous structure. This conical shape suggests that trajectories start from a similar region and then diverge according to well-structured dynamic rules, consistent with the hypothesis that the observed behavior has an intrinsic low dimensionality, well represented in 2D but embedded in a broader dynamic space.

Panel (C) displays the trajectories in the space of time-integrated control inputs ($\int c_1, \int c_2$). The

clear separation between conditions indicates that the latent dynamics are not only structured but also discriminative. The fact that the trajectories are well-separated and structured suggests that the latent dynamics captured by MDDS are consistent with the experimental conditions, offering sharper separation compared to static methods like PCA or UMAP.
Panels (D) and (E) present the temporal evolution of the integrated controls $\int c_1(t)$ and $\int c_2(t)$, respectively. The distinct time courses further confirm that the inferred dynamics are condition-specific and evolve consistently across trials.
Panel (F) shows the reconstructed time series of selected observed variables, demonstrating that the learned latent dynamics successfully capture the main features of the original data.
Finally, Panels (G-I) provide a dynamical analysis of the system through the estimated Jacobians, Schur decompositions, and eigenvalue spectra of the learned dynamical system. The eigenvalue distributions suggest a mixture of stable and weakly unstable modes, consistent with a dynamical system that supports rich yet structured temporal evolution.

3.3 UMAP Hyperparameter Search

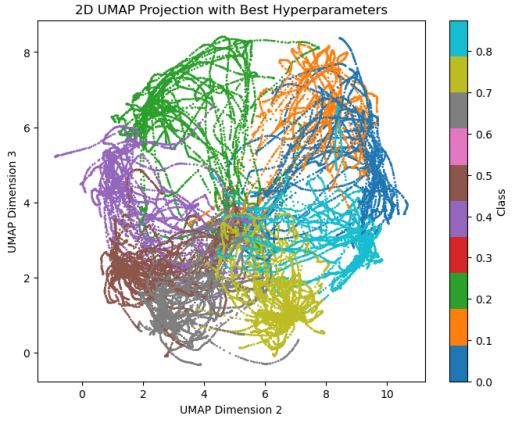


Figure 8: UMAP Embedding of the neural data after hyperparameter optimization with Optuna.

The optimal hyperparameters for UMAP embedding identified after 20 trials in Optuna are `n_neighbors`= 24, `min_dist`= 0.129, `n_components`=9. While Optuna's hyperparameter search does not guarantee finding globally optimal values, it successfully identified parameters that yield a meaningful representation of the data. The resulting UMAP embedding is depicted in Figure 8, demonstrating its ability to separate the 8 target angles despite having no information about them during the embedding process.

3.4 MDDS can denoise neural data for Classification tasks

We evaluated an LDA classifier's ability to decode the target angle using neural activity representations at the final time point of the monkey's reach. Successful decoding ($\rho_M > 0$) indicates that neural activity at this stage encodes the reached target angle. The LDA was trained separately on the representations generated using various dimensionality reduction techniques (Raw, PCA, UMAP, MDDS), as detailed in Methods. Comparing classification performances across these methods reveals which representation is most suitable for making the encoded target angle information linearly separable for the classification task.

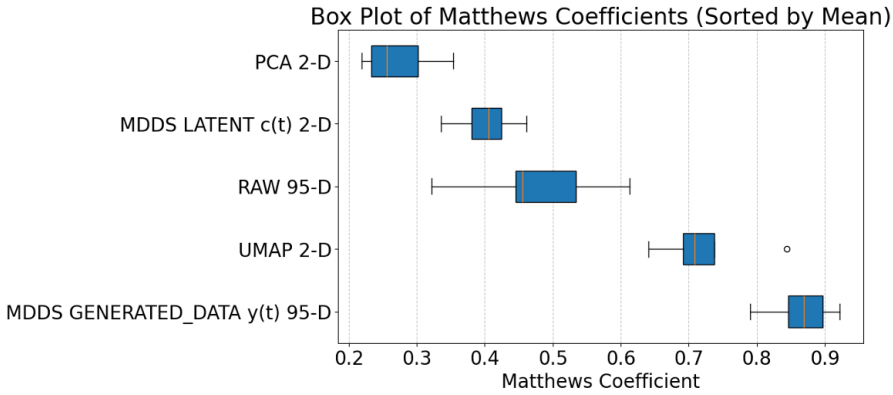


Figure 9: Box plot of Matthews Correlation Coefficients: LDA classification of the target angle using different neural data representations (RAW, PCA, UMAP, MDDS latent, and MDDS generated data) at the final time point. The results are sorted by the mean coefficient.

The classification results presented in Table 1 and Figure 9 shows that, for the same number of retained dimensions (95-D), the MDDS-generated representation consistently outperforms the raw neural data, as indicated by the higher Matthews correlation coefficient ($\rho_M \approx 0.87$ for MDDS against $\rho_M \approx 0.47$ for RAW data). This result suggests that the MDDS method effectively enhances the linear separability of the neural data, due to the latent-model representation.

The best low-dimensional (2-D) representation of neural data for linear classification of target angle using the last time point was achieved by an optimized UMAP ($\rho_M \approx 0.74$).

Using techniques that reveal non-linear manifolds (MDDS and UMAP) improves classification performance even using the same number of dimensions. This suggests that the manifold for discriminating motor commands used in this task, in motor cortex, is curved. The task-relevant neural activity crucial for decoding is likely evolving on a non-linear curved manifold.

Table 1: Classification results using different neural data representation. Rows are sorted first by the number of retained dimensions and then by classification performance.

Neural representation type	**Matthews Coefficient,** $\rho_M \in [-1, 1]$	**Std Matthews Coeffs**
MDDS Generated data $y(t)$ 95-D	**0.865108**	0.0452809
RAW Neural data 95-D	0.473754	0.0972355
UMAP 2-D	**0.740427**	0.0625496
MDDS Latent trajectories $\int c(\tau)d\tau$ 2-D	0.401399	0.0420250
PCA 2-D	0.272907	0.0493123

3.5 Regression of kinematic data using linear regression

Linear regression of primary motor cortex (M1) neuronal activity to kinematic parameters (i.e. position) is of interest for both Brain-Computer Interfaces (BCI) and general Neuroscience. It is important to study how different motor commands and movement parameters are represented in neuronal population activity[1] [9]. For BCI application, the ability to decode kinematic intent from the neural

signal is crucial for enabling prosthetic control or cursor movement on a screen [8] [6] [12] [14]. Table 2 reports the results for linear regression of kinematic data

Table 2: R^2 Scores (Mean and Std) using different neural data dimensionality reduction techniques.

#	**Name**	**R^2 Mean**	**R^2 Std**
1	MDDS Generated data $y(t)$ 95-D	0.662667	0.00668309
2	RAW Neural data 95-D	0.565325	0.00737082
3	PCA 3-D	0.148393	0.00758670
4	MDDS Latent trajectories $\int c(\tau)d\tau$ 2-D	0.120509	0.00469726
5	UMAP 2	0.106768	0.00207547

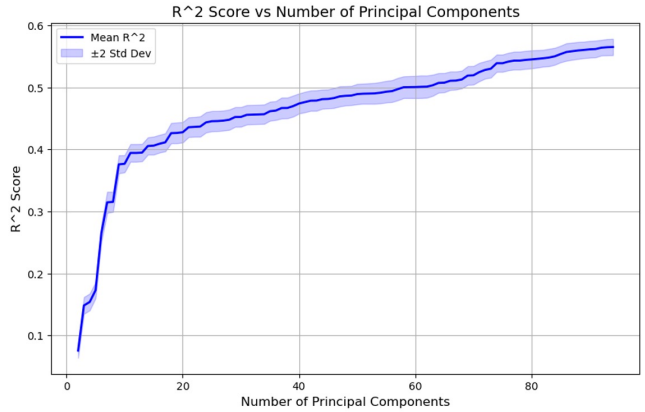


Figure 10: Performances on the regression of kinematic(R^2-score varying the number of principal components used in input.

In Figure 10 we observe how well the cursor (x,y) position can be approximated with linear regression as a function of the number of principal components (PCs) of the neural data. The R^2-score, the percentage of variance in kinematics explained by the regression, rises very steeply initially with increasing numbers of PCs added. But after some 20-30 principal components, the curve begins to flatten, approaching as expected the same performances of using the RAW neural data 95-D (see Table2). The presence of such a plateau means that adding more than a certain number of principal components does not significantly improve the linear decoding of kinematics variable. This confirms the findings in literature that neural activity in M1 relies on a manifold with a dimensionality lower than the number of recorded units [9] [6] [1].

The best-performing regressor uses MDDS-generated 95-D data, outperforming raw neural data of the same dimensionality in linear regression of kinematic variables. This suggests MDDS effectively denoises neural signals, retaining only task-relevant information of the trajectory.

3.6 Brain-Computer Interface Model

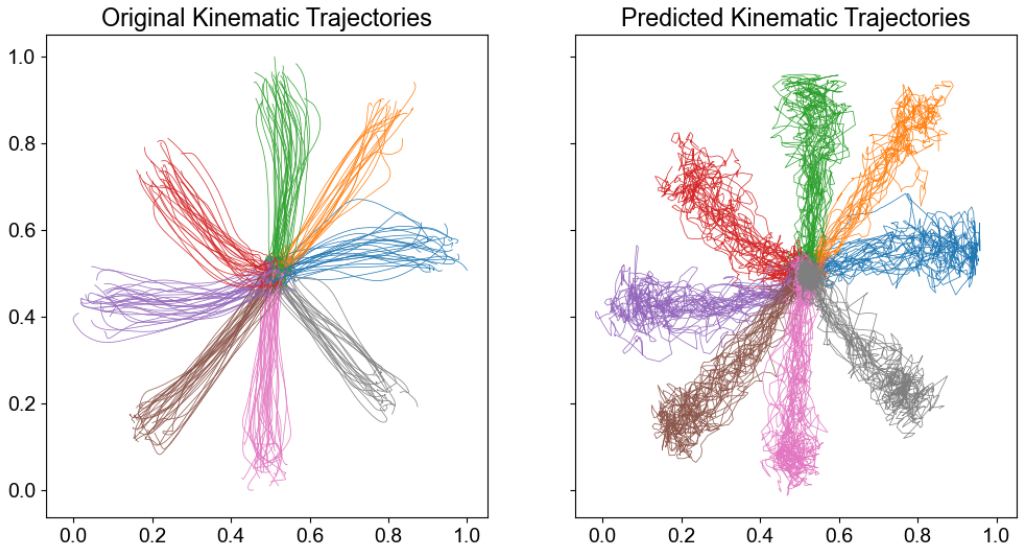


Figure 11: A comparison of original and predicted kinematic trajectories utilizing our BCI model.

The results of the BCI model described in Section 2.9 are presented in Figure 11. As illustrated in this figure, the predicted kinematic trajectories exhibit a chaotic nature and do not closely align with the original behavior of the data. Despite this discrepancy, the employment of a non-linear feed-forward decoder enables the mapping of these trajectories into a space that bears resemblance to the original one.

This approach highlights the challenges in accurately predicting complex behaviors using current BCI methodologies while also showcasing the potential of non-linear decoders in approximating such dynamics.

4 Conclusions

In this study, we demonstrate that motor cortex population activity during a center-out reach task evolves on a low-dimensional, curved manifold. Applying non-linear dimensionality reduction (MDDS) yielded a cone-like manifold that both denoised the neural signals and let the target direction information more linearly separable —outperforming PCA, UMAP, and raw data in LDA classification ($\rho_M \approx 0.87$(MDDS) vs. $\rho_M \approx 0.47$(Raw neural data) at 95-D). Similarly, MDDS-generated representations improved linear regression of joystick kinematics ($R^2 \approx 0.66$ (MDDS) vs. $R^2 \approx 0.57$ (Raw neural data)). Together, these results provide evidence that MDDS is a good dimensionality technique for retaining task-relevant information. Furthermore, neural dynamics relevant for both decoding discrete targets and continuous movement variables are constrained to a curved manifold rather than a flat subspace, see comparison between PCA and MDDS performances in Table 1, 2.

We further show that a simple Linear Dynamical System trained on MDDS-denoised trajectories can recapitulate the observed neural trajectories, suggesting a minimal computational model in which preparatory activity initializes a point on the manifold and subsequent autonomous dynamics drive movement generation. For real-time Brain-Computer Interface applications, this implies that lever-

aging manifold-based latent factors can enhance both classification- and regression-based decoders, improving robustness and accuracy under linear readout constraints.

In this study, we investigated whether neural population activity recorded during a motor task exhibits condition-specific low-dimensional dynamics that reflect the structure of behavior. Through a combination of dimensionality reduction techniques and dynamical systems modeling, we provide converging evidence that distinct experimental conditions evoke differentiable neural trajectories embedded within a structured low-dimensional manifold.

Initial qualitative analysis of kinematic data revealed that different task conditions reliably elicited distinct movement patterns. This behavioral differentiation was mirrored at the neural level: both PCA and UMAP uncovered condition-dependent structure in neural activity, with UMAP offering greater separability than linear methods. However, only the application of MDDS enabled explicit modeling of the underlying neural dynamics, revealing interpretable latent trajectories with distinct temporal signatures for each condition. Notably, the trajectories inferred by MDDS originated from a common initial state and diverged systematically, forming a cone-like geometry suggestive of structured control processes.

These findings support the view that motor cortical activity is governed by condition-specific low-dimensional dynamics, which can be uncovered and characterized using nonlinear dynamical systems approaches. The MDDS framework, in particular, offers a promising avenue for extracting interpretable and behaviorally relevant structure from neural time series.

References

[1] Mehdi Aghagolzadeh and Wilson Truccolo. Inference and decoding of motor cortex low-dimensional dynamics via latent state-space models. 24(2):272–282.

[2] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus A. F. Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 05 2000.

[3] Mark M. Churchland, Byron M. Yu, John P. Cunningham, Leo P. Sugrue, Marlene R. Cohen, Greg S. Corrado, William T. Newsome, Andrew M. Clark, Paymon Hosseini, Benjamin B. Scott, David C. Bradley, Matthew A. Smith, Adam Kohn, J. Anthony Movshon, Katherine M. Armstrong, Tirin Moore, Steve W. Chang, Lawrence H. Snyder, Stephen G. Lisberger, Nicholas J. Priebe, Ian M. Finn, David Ferster, Stephen I. Ryu, Gopal Santhanam, Maneesh Sahani, and Krishna V. Shenoy. Stimulus onset quenches neural variability: a widespread cortical phenomenon. 13(3):369–378. Publisher: Nature Publishing Group.

[4] John P. Cunningham and Byron M. Yu. Dimensionality reduction for large-scale neural recordings. 17(11):1500–1509. Publisher: Nat Neurosci.

[5] Cátia Fortunato, Jorge Bennasar-Vázquez, Junchol Park, Joanna C. Chang, Lee E. Miller, Joshua T. Dudman, Matthew G. Perich, and Juan A. Gallego. Nonlinear manifolds underlie neural population activity during behaviour. page 2023.07.18.549575.

[6] Juan A. Gallego, Matthew G. Perich, Lee E. Miller, and Sara A. Solla. Neural manifolds for the control of movement. 94(5):978–984. Publisher: Cell Press.

[7] Juan A. Gallego, Matthew G. Perich, Stephanie N. Naufel, Christian Ethier, Sara A. Solla, and Lee E. Miller. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. 9(1):4233. Publisher: Nature Publishing Group.

[8] Leigh R. Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y. Masse, John D. Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S. Cash, Patrick van der Smagt, and John P. Donoghue. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. 485(7398):372–375, 2012.

[9] Svenja Melbaum, Eleonora Russo, David Eriksson, Artur Schneider, Daniel Durstewitz, Thomas Brox, and Ilka Diester. Conserved structures of neural activity in sensorimotor cortex of freely moving rats allow cross-subject decoding. 13(1):1–14. Publisher: Nature Publishing Group.

[10] Emily R. Oby, Alan D. Degenhart, Erinn M. Grigsby, Asma Motiwala, Nicole T. McClain, Patrick J. Marino, Byron M. Yu, and Aaron P. Batista. Dynamical constraints on neural population activity. 28(2):383–393. Publisher: Nature Publishing Group.

[11] Matthew G. Perich, Juan A. Gallego, and Lee E. Miller. A neural population mechanism for rapid learning. *Neuron*, 100(4):964–976, 2018.

[12] J. Andrew Pruszynski and Jörn Diedrichsen. Reading the mind to move the body decoding neural signals of intention and movement should guide the development of neural prosthetics. 348(6237):860–861. Publisher: American Association for the Advancement of Science.

[13] 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. 512(7515):423–426. Publisher: Nature.

[14] Sergey D. Stavisky, Jonathan C. Kao, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. A high performing brain-machine interface driven by low-frequency local field potentials alone and together with spikes. 12(3). Publisher: J Neural Eng.

Linear manifold-generating dynamical systems

Simon Altrogge[1], Ben Griffin[2], Alice Marraffa[3], Christian Müller[4]

Abstract

We investigate the properties of linear manifold-generating dynamical systems, a class of dynamical systems whose trajectories are constrained to evolve on lower-dimensional submanifolds that foliate the phase space.

In such systems, the dynamics of each trajectory are governed by a time-dependent linear combination of m mutually commuting linear vector fields. This structure ensures the integrability of the foliation structure and enables a compact exponential closed-form solution for the system's evolution. As a result, there is a significant computational advantage: direct evaluation of system states at arbitrary times, bypassing costly numerical integration. This approach significantly improves computational efficiency in data-driven modeling, particularly in high-dimensional settings.

We extend the framework by incorporating nonhomogeneous linear terms under specific algebraic constraints and show that the manifold-generating structure is preserved under smooth, invertible, component-wise coordinate transformations.

1 Introduction

The behavior of many biological systems can be described well using the theory of dynamical systems. In neuroscience, for example, the dynamics of single neurons as well as entire networks of neurons are frequently modeled as dynamical systems [1, 2]. Moreoreover, the dimensionality of neural networks, as of many other biological systems, is quite high. According to the manifold hypothesis [3], high-dimensional data tends to be constrained to the vicinity of low-dimensional manifolds, a view that has become increasingly widespread also in neuroscience [4]. If such high-dimensional data is obtained from a high-dimensional dynamical system, the structure of the corresponding low-dimensional manifold must somehow be encoded in the properties of the underlying dynamical system. In this report, we will investigate this very relationship.

A dynamical system can quite generally be described by its state $\mathbf{x}(t) \in \mathbb{R}^n$ at time t and a, possibly time-dependent, vector field $\mathbf{f}(t, \mathbf{x}(t)) : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ that describes the evolution of said state in time,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}(t, \mathbf{x}(t)). \tag{1}$$

Consequently, if we initialize the dynamical system to a certain state $\mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbb{R}^n$ at some time t_0 it will evolve according to eq. (1) and with that trace a trajectory $\{\mathbf{x}(t)|\mathbf{x}(t_0) = \mathbf{x}_0, t \geq t_0\}$ in state space. In this report we will now study a specific subclass of dynamical systems, namely, linear manifold-generating dynamical systems.

To not get ahead of ourselves, we will introduce these two properties one by one, starting with linearity. A dynamical system is called *linear* if its defining vector field is linear. Such a linear vector

[1]Neural Network Dynamics and Computation, Institute of Genetics, University of Bonn, Bonn, Germany
[2]Oxford Centre for Functional MRI of the Brain, Oxford Centre for Integrative Neuroimaging, University of Oxford, Oxford, UK
[3]DMATH, ETH Zurich, Zurich, Switzerland
[4]INM - Leibniz Institute for New Materials, Saarbrücken, Germany

field $\mathbf{W}(t, \mathbf{x}(t))$ can be expressed in terms of a matrix $W(t) \in \mathbb{R}^{n\times n}$ and acts on the state $\mathbf{x}(t)$ by matrix multiplication,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{W}(t, \mathbf{x}(t)) = W(t)\mathbf{x}(t). \tag{2}$$

A dynamical system is called *manifold-generating* if all its trajectories which pass through a distinguished point in state space, $\mathbf{x}_0 \in \mathbb{R}^n$, lie on an m-dimensional submanifold $\mathcal{M} \subset \mathbb{R}^n$,

$$\{\mathbf{x}(t) | \exists t_0 : \mathbf{x}(t_0) = \mathbf{x}_0\} \subseteq \mathcal{M}.$$

Crucially, every dynamical system that generates an m-dimensional manifold can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \sum_{i=1}^{m} c_i(t)\mathbf{f}_i(\mathbf{x}), \tag{3}$$

that is, the dynamics of the state $\mathbf{x}(t)$ are determined by a superposition of time-independent vector fields $\mathbf{f}_1, \ldots, \mathbf{f}_m : \mathbb{R}^n \to \mathbb{R}^n$ with time-dependent *controls* $c_1, \ldots, c_m : \mathbb{R} \to \mathbb{R}$. Being of the form presented in eq. (3) is, however, only necessary but not yet sufficient to be manifold-generating. According to the Frobenius theorem [5], to be manifold-generating the dynamical system additionally has to satisfy the Frobenius condition which requires that the Lie brackets of its vector fields $\mathbf{f}_1, \ldots \mathbf{f}_m$ lie again within the span of said vector fields,

$$[\mathbf{f}_i, \mathbf{f}_j](\mathbf{x}) = J_{\mathbf{f}_j}(\mathbf{x})\mathbf{f}_i(\mathbf{x}) - J_{\mathbf{f}_i}(\mathbf{x})\mathbf{f}_j(\mathbf{x}) \in \mathrm{span}\{\mathbf{f}_k(\mathbf{x})\}_{k\in[m]}. \tag{4}$$

Here, $J_{\mathbf{f}_i}(\mathbf{x}) \in \mathbb{R}^{n\times n}$ is the Jacobian matrix of the vector field $\mathbf{f}_i$ at the point $\mathbf{x}$ in state space.

We can now combine the two notions, that is, eqs. (2) and (3), to arrive at dynamical systems of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \sum_{i=1}^{m} c_i(t)\mathbf{W}_i(\mathbf{x}(t)) = \sum_{i=1}^{m} c_i(t)W_i\mathbf{x}(t). \tag{5}$$

Thus, the dynamical system is given by a set of time-independent linear vector fields $\mathbf{W}_1, \ldots, \mathbf{W}_m$ with $\mathbf{W}_i(\mathbf{x}(t)) = W_i\mathbf{x}(t)$. For such a system, the Frobenius condition of eq. (4) can be rewritten as

$$\begin{aligned}[\mathbf{W}_i, \mathbf{W}_j](\mathbf{x}) &= J_{\mathbf{W}_j}(\mathbf{x})\mathbf{W}_i(\mathbf{x}) - J_{\mathbf{W}_i}(\mathbf{x})\mathbf{W}_j(\mathbf{x}) \\ &= W_jW_i\mathbf{x} - W_iW_j\mathbf{x} \in \mathrm{span}\{W_k\mathbf{x}\}_{k\in[m]}.\end{aligned} \tag{6}$$

Dynamical systems of the form presented in eq. (5) that satisfy the Frobenius condition of eq. (6) are referred to as *linear manifold-generating* dynamical systems. It is the theory of and numerical methods for these linear manifold-generating linear systems that the remainder of this report is concerned with.

2 Results

Simultaneously diagonalizable vector fields give rise to linear manifold-generating dynamical systems

According to the Frobenius theorem, a linear dynamical system is manifold-generating if the Lie brackets of its vector fields $\mathbf{W}_1, \ldots, \mathbf{W}_m$ lie again within the span of these vector fields (eq. (6)). As

each vector field $\mathbf{W}_i$ is linear (eq. (5)), it can be expressed in terms of a matrix W_i. If we now assume all these matrices to commute, that is, $W_jW_i = W_iW_j$, then

$$\begin{aligned}[\mathbf{W}_i, \mathbf{W}_j](\mathbf{x}) &= W_jW_i\mathbf{x} - W_iW_j\mathbf{x} \\ &= (W_jW_i - W_iW_j)\,\mathbf{x} \\ &= (W_iW_j - W_iW_j)\,\mathbf{x} \\ &= 0\mathbf{x} \\ &= \mathbf{0}.\end{aligned}$$

For linear vector fields with commuting matrices, the Lie brackets vanish and, thus, trivially lie within the span of the vector fields, as $\mathbf{0} \in \operatorname{span}\{\mathbf{W}_i\}_{i\in[m]}$. Linear dynamical systems of commuting matrices are therefore manifold-generating.

This is in particular true for linear dynamical systems given by a set of vector fields $\{\mathbf{W}_i\}_{i\in[m]}$ of simultaneously diagonalizable matrices, $W_i = V\operatorname{diag}(\mathbf{s}_i)V^{-1}\ \forall i \in [m]$. Such matrices have the same eigenvectors $\mathbf{v}_j = V_{*j} \in \mathbb{R}^n$ for $j \in [n]$ with respective eigenvalues $(\mathbf{s}_i)_j$ and, notably, commute:

$$\begin{aligned}W_jW_i &= V\operatorname{diag}(\mathbf{s}_j)V^{-1}V\operatorname{diag}(\mathbf{s}_i)V^{-1} \\ &= V\operatorname{diag}(\mathbf{s}_j)\operatorname{diag}(\mathbf{s}_i)V^{-1} \\ &= V\operatorname{diag}(\mathbf{s}_i)\operatorname{diag}(\mathbf{s}_j)V^{-1} \\ &= V\operatorname{diag}(\mathbf{s}_i)V^{-1}V\operatorname{diag}(\mathbf{s}_j)V^{-1} \\ &= W_iW_j.\end{aligned}$$

Here we use $V^{-1}V = \mathbb{1}$ and, to proceed from the second to third line, that all diagonal matrices commute with one another. Going forward we will restrict ourselves to such linear manifold-generating dynamical systems of simultaneously diagonalizable matrices as they turn out to have useful properties, which we can exploit.

There exists a simple expression for solutions of linear manifold-generating dynamical systems of simultaneously diagonalizable vector fields

One of these properties is that the solutions of linear manifold-generating dynamical systems of commutative, and in particular simultaneously diagonalizable, matrices take the simple form

$$\mathbf{x}(t) = \exp\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau\right)\mathbf{x}_0 \tag{7}$$

given some initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. The proof goes as follows:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) &= \frac{\mathrm{d}}{\mathrm{d}t}\exp\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau\right)\mathbf{x}_0 = \frac{\mathrm{d}}{\mathrm{d}t}\sum_{n=0}^{\infty}\frac{\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau\right)^n}{n!}\mathbf{x}_0 = \sum_{n=0}^{\infty}\frac{\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau\right)^n}{n!}\mathbf{x}_0 \\ &= \sum_{n=1}^{\infty}\frac{1}{n!}\left[\sum_{k=0}^{n-1}\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau\right)^k W(t)\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau\right)^{n-1-k}\right]\mathbf{x}_0\end{aligned}$$

by the product rule. For the next step, we have to show that $W(t)$ commutes with its integral $\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau$. For the specific dynamical systems we consider here we have $W(t) = \sum_{i=1}^{m} c_i(t) W_i$ with $W_i W_j = W_j W_i$, so

$$\begin{aligned}
\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau\, W(t) &= \sum_{i=1}^{m} \int_{t_0}^{t} c_i(\tau)\,\mathrm{d}\tau\, W_i \sum_{j=1}^{m} c_j(t) W_j = \sum_{i,j=1}^{m} \int_{t_0}^{t} c_i(\tau)\,\mathrm{d}\tau\, c_j(t) W_i W_j \\
&= \sum_{i,j=1}^{m} \int_{t_0}^{t} c_i(\tau)\,\mathrm{d}\tau\, c_j(t) W_j W_i = \sum_{j=1}^{m} c_j(t) W_j \sum_{i=1}^{m} \int_{t_0}^{t} c_i(\tau)\,\mathrm{d}\tau\, W_i \\
&= W(t) \int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau.
\end{aligned}$$

With that, continuing the proof from above, we obtain

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) &= \sum_{n=1}^{\infty} \frac{1}{n!} \left[\sum_{k=0}^{n-1} W(t) \left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau \right)^{n-1} \right] \mathbf{x}_0 = \sum_{n=1}^{\infty} \frac{1}{(n-1)!} W(t) \left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau \right)^{n-1} \mathbf{x}_0 \\
&= W(t) \sum_{m=0}^{\infty} \frac{\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau \right)^{m}}{m!} \mathbf{x}_0 = W(t) \exp\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau \right) \mathbf{x}_0 \\
&= W(t)\mathbf{x}(t),
\end{aligned}$$

proving that eq. (7) is indeed a solution of the differential equation describing a linear dynamical system (eq. (2)).

Linear manifold-generating dynamical systems of simultaneously diagonalizable vector fields can be evaluated efficiently

In the special case of simultanously diagonalizable matrices, we can rewrite eq. (7) as

$$\begin{aligned}
\mathbf{x}(t) = \exp\left(\int_{t_0}^{t} W(\tau)\,\mathrm{d}\tau \right) \mathbf{x}_0 &= \exp\left(\int_{t_0}^{t} \sum_{i=1}^{m} c_i(\tau) W_i\,\mathrm{d}\tau \right) \mathbf{x}_0 \\
&= \exp\left(\sum_{i=1}^{m} W_i \int_{t_0}^{t} c_i(\tau)\,\mathrm{d}\tau \right) \mathbf{x}_0 \\
{\scriptstyle C_i(t) := \int_{t_0}^{t} c_i(\tau)\,\mathrm{d}\tau \rightsquigarrow} \quad &= \left[\prod_{i=1}^{m} \exp\left(V\,\mathrm{diag}(\mathbf{s}_i) V^{-1} C_i(t) \right) \right] \mathbf{x}_0 \\
&= \left[\prod_{i=1}^{m} \exp\left(V\,\mathrm{diag}\left(\mathbf{s}_i C_i(t) \right) V^{-1} \right) \right] \mathbf{x}_0 \\
&= V \left[\prod_{i=1}^{m} \mathrm{diag}\left(\exp\left(\mathbf{s}_i C_i(t) \right) \right) \right] V^{-1} \mathbf{x}_0 \\
&= V \left[\prod_{i=1}^{m} \mathrm{diag}\left(\exp\left(\mathbf{s}_i \right)^{C_i(t)} \right) \right] V^{-1} \mathbf{x}_0.
\end{aligned}$$

Rearranging the exponential of a sum of m matrices in line 2 into a product of m matrix exponentials as in line 3 is only possible here because all matrices W_i within the sum commute by assumption. Finally, the simultaneous diagonalizability allows us to simplify the matrix exponential considerably.

Notably, if it is not a single but multiple, say K, trajectories we need to evaluate, we can reuse large parts of the calculations necessary for one trajectory for all the others as they only differ in the controls $\left\{c_i^{(k)}(t)\right\}_{k\in[K]}$ and, thus, their integrals $\left\{C_i^{(k)}(t)\right\}_{k\in[K]}$.

Closed-form evaluation enables direct trajectory access

Before contrasting analytic and numerical approaches in the linear manifold-generating setting, we briefly recall Euler's method, one of the simplest numerical schemes for solving ordinary differential equations.

Given a system of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)),$$

Euler's method approximates the trajectory by iteratively updating:

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \cdot f(\mathbf{x}(t)),$$

where Δt is a small time step. This produces a piecewise-linear approximation to the solution over a time window T.

In the context of linear dynamical systems, where $f(\mathbf{x}) = W(t)\mathbf{x}$, this becomes:

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \cdot W(t)\mathbf{x}(t).$$

While easy to implement, this method accumulates error over time and becomes inefficient when solving systems repeatedly or at large scale.

As we have seen, when all matrices W_i in the decomposition

$$W(t) = \sum_{i=1}^{m} c_i(t) W_i$$

commute (e.g., when they are simultaneously diagonalizable), the system admits a closed-form solution:

$$\mathbf{x}(t) = \left(\prod_{i=1}^{m} \exp\left(\left(\int_0^t c_i(\tau)\, d\tau \right) W_i \right) \right) \mathbf{x}_0.$$

This avoids the need for numerical integration and yields an exact expression that is both efficient and interpretable. Since the solution depends only on integrals of the control functions up to time t, it allows direct evaluation of $\mathbf{x}(t)$ without stepping through all intermediate times $0, \Delta t, 2\Delta t, \ldots, t - \Delta t$.

This idea is illustrated in Fig. 1. Instead of incrementally stepping through intermediate timepoints as required by numerical schemes like Euler's method, the closed-form solution enables us to directly "leap" to any desired timepoint. This is particularly advantageous when the system's internal dynamics unfold faster than measurements are recorded, or when data is sparsely sampled in time.

This efficiency becomes even more valuable when evaluating many trajectories in parallel. For instance, if each trajectory is governed by its own control functions $c_i^{(k)}(t)$, we have:

$$\dot{\mathbf{x}}^{(k)}(t) = \sum_{i=1}^{m} c_i^{(k)}(t) W_i \mathbf{x}^{(k)}(t), \quad \mathbf{x}^{(k)}(0) = \mathbf{x}_0^{(k)},$$

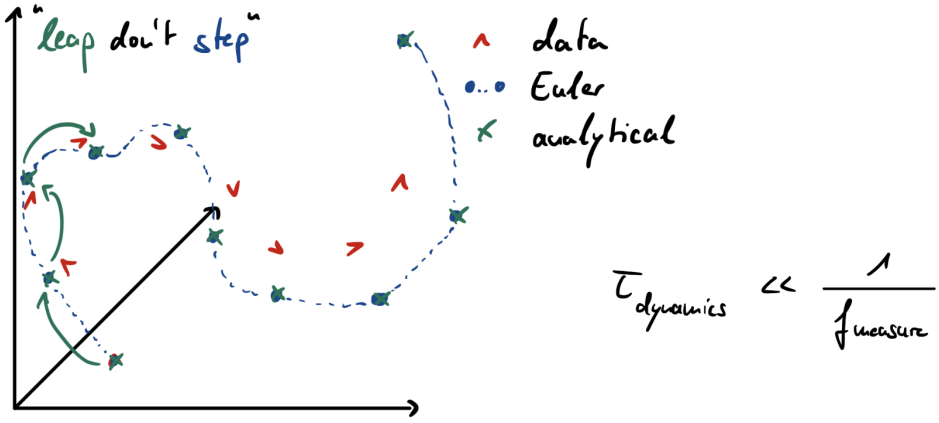


Figure 1: **"Leap don't step":** Illustration of the difference between numerical integration and closed-form evaluation. Euler integration (blue dotted path) requires sequential stepping through intermediate timepoints, whereas closed-form solutions (green arrows) allow direct jumps to desired times. This is especially useful when dynamics evolve more rapidly than measurements are sampled, or when only a few discrete timepoints are available.

which Euler would approximate as:

$$\mathbf{x}^{(k)}(t + \Delta t) \approx \mathbf{x}^{(k)}(t) + \Delta t \cdot W^{(k)}(t)\mathbf{x}^{(k)}(t),$$

but for which we can write an exact solution:

$$\mathbf{x}^{(k)}(t) = \left(\prod_{i=1}^{m} \exp\left(\left(\int_0^t c_i^{(k)}(\tau)\, d\tau \right) W_i \right) \right) \mathbf{x}_0^{(k)}.$$

This formulation enables accurate, closed-form evaluation of an entire family of trajectories without requiring time discretisation.

Analytical solutions offer substantial runtime gains

We next empirically compare the computational efficiency of the closed-form solution against Euler integration, using three variants:

- **Euler:** classic step-based numerical integration,
- **Analytical (for-loop):** analytical solution evaluated independently for each trajectory using a for-loop,
- **Analytical (vectorised):** fully vectorised implementation that evaluates all trajectories in parallel.

As shown in Fig. 2, the numerical and analytical approaches produce consistent simulated trajectories, validating the correctness of the analytical formulation. However, Fig. 3 reveals clear differences in runtime. Across all scenarios (varying system dimensionality, number of W_i matrices, and number of time steps), the analytical methods scale more favourably than the numerical approach. Even

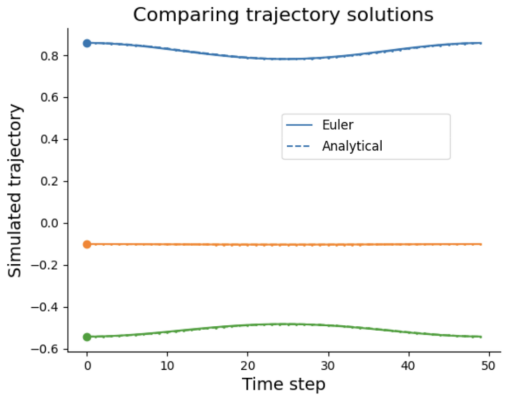


Figure 2: **Trajectory comparison:** Three example trajectories (indicated by color), shown using two methods: Euler integration (solid lines) and analytical evaluation (dashed lines). The close agreement between line styles demonstrates consistency between the numerical and analytical solutions.

the for-loop implementation is consistently faster than Euler integration, with the vectorised version offering the greatest speedup.

These benchmarks highlight the benefits of the closed-form approach not only for accuracy, but also for computational efficiency; especially in high-dimensional, multi-trajectory, or long-horizon settings.

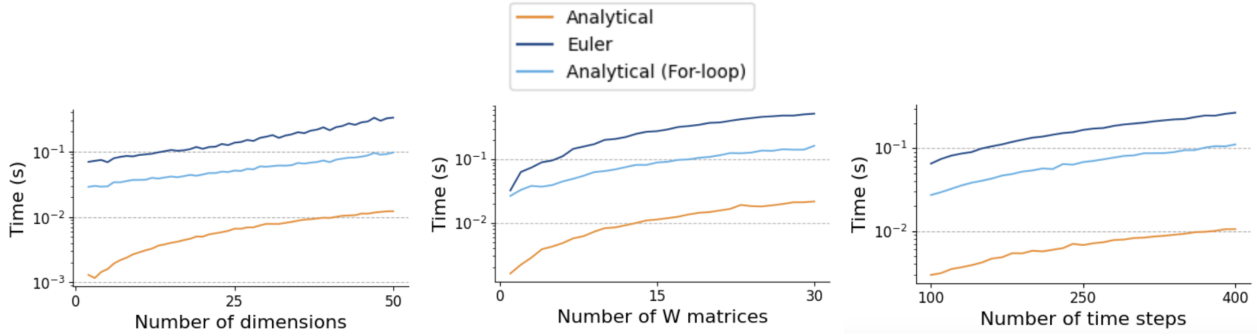


Figure 3: **Computational efficiency of Euler vs. analytical solutions.** Left: Runtime vs. system dimensionality. Middle: Runtime vs. number of W_i matrices. Right: Runtime vs. number of timesteps. Euler becomes increasingly expensive with longer time horizons, while the vectorised analytical method remains nearly constant.

Fitting the analytical model parameters to random trajectories

We characterize the model's performance when applied to artificially generated data. For this purpose, the initial conditions $\mathbf{x}_0$, V, s_i and the $c_i^{(k)}(t)$ coefficients are chosen randomly. Using these inputs, the reference trajectories are created from 2 by an Euler algorithm, where the r.h.s. of 2 is augmented with a mild amount of random noise.

Since we want to fix the the V matrix as orthogonal, we do not assume all of its component to be fit parameters. Instead, we apply rotations to the unit matrix, where an $N \times N$ matrix can undergo

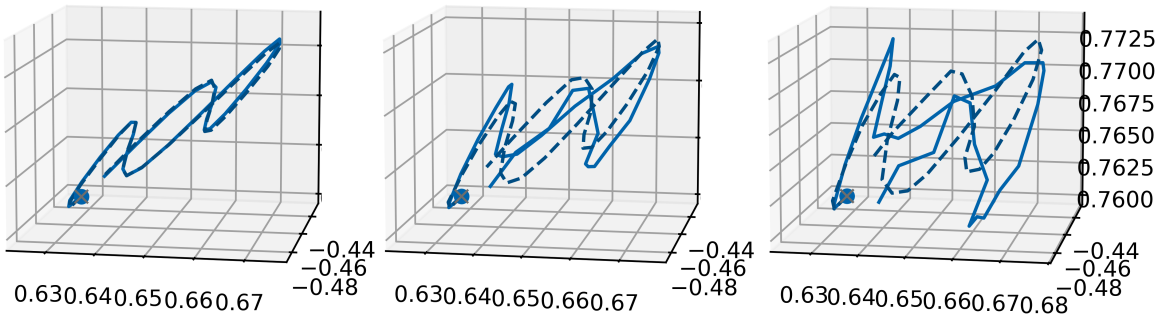


Figure 4: Example reference trajectory (solid line) and fitted model trajectory (dashed line) with noise level on the reference increasing from panel (a) to (b) to (c). The overall shape of the fitted trajectory remains relatively stable against the noise.

rotations in $\alpha \in [1, n(n-1)/2]$ different planes. We describe each of these rotations by the tangent of the angle $\tan(\varphi_\alpha) \in (-\infty, \infty)$, which is more robust in numerical optimization than introducing periodicity with respect to the φ_α. The loss function L for the optimization process therefore reads

$$F(\mathbf{x}_0, s_i, \tan(\varphi_\alpha), c_i^{(k)}(t)) = V \left(\prod_{i=1}^{m} \exp(\mathrm{diag}(\mathbf{s}_i))^{\int_0^t c_i^{(k)}(\tau)\mathrm{d}\tau} \right) V^{-1} \mathbf{x}_0$$

$$L(\mathbf{x}_0, s_i, \tan(\varphi_\alpha), c_i^{(k)}(t)) = \sum_k \sum_t \left| \mathbf{x}(t)^{(k)} - F(\mathbf{x}_0, s_i, \tan(\varphi_\alpha), c_i^{(k)}(t)) \right|^2, \tag{8}$$

and is minimized with respect to all of its explicitly stated arguments, except $\mathbf{x}_0$, which is assumed to be known. At the start of an optimization, all parameters are initiated randomly.

To get an intrinsic estimate, the loss function can also be reformulated using the arithmetic mean instead of just the sum in 8. Both approaches were tried and yielded similar results.

Manifold-generating nonhomogeneous linear dynamical systems

Up to this point, our analysis has focused on a specific subclass of systems whose trajectories are constrained to evolve with nonautonomous linear dynamics on a family of m-dimensional submanifolds foliating the phase space. As established in previous sections, this class of systems is particularly convenient since it admits a straightforward criterion to determine the manifold-generating property. Specifically, the matrices W_i defining the m linear vector fields $W_i\mathbf{x}$ in Eq. 5 have to mutually commute. Furthermore, such systems possess an analytical solution (see Equation 7) that can be used to speed up the optimization process when fitting evolution rules of trajectories to empirical data.

However, this setting is somewhat restrictive, particularly when attempting to model real-world systems, which often exhibit more complex behavior. We are therefore motivated to explore whether it is possible to broaden this class of systems while still preserving the manifold-generating property, possibly by introducing additional, yet still simple, constraints on their vector fields.

Here, we focus on nonautonomous, nonhomogeneous linear dynamical systems defined by adding some bias vectors b_i, $i = 1, \dots M$ to the linear vector fields $W_i\mathbf{x}$

$$\dot{x} = \sum_{i=1}^{m} c_i(t)\big(W_i\mathbf{x} + \mathbf{b}_i\big) \tag{9}$$

Since we want to maintain the manifold-generating property, a sufficient condition is again required, for all $i, j = 1, \ldots, m$,

$$[\mathbf{f}_i, \mathbf{f}_j] = [W_i\mathbf{x} + \mathbf{b}_i, W_j\mathbf{x} + \mathbf{b}_j] = \mathbf{0}$$

This requirement can be satisfied by imposing the following

- the matrices W_i still commute: $W_iW_j = W_jW_i$
- each bias $\mathbf{b}_i$ is an eigenvector of each W_j, $i \neq j$, with eigenvalue zero: $W_i\mathbf{b}_j = 0$

The proof of this fact can be found in Appendix A.

Given an initial condition $\mathbf{x}(0) = \mathbf{x}_0$, the solution to the initial value problem can again be put in a simple exponential form

$$\mathbf{x}(t) = \left(\prod_{i=1}^{m} \exp\left(\left(\int_0^t c_i(\tau)d\tau \right) W_i \right) \right) \mathbf{x}_0 + \sum_{i=1}^{m} \left(\int_0^t c_i(\tau)d\tau \right) \mathbf{b}_i \tag{10}$$

Nonlinearly Transformed Manifold-Generating Linear Systems

We now examine the impact of a nonlinear change of coordinates on a manifold-generating linear system. Although these transformations do not alter the nature of the dynamics and the consequent linear behavior of the trajectories, they are relevant when considering changes in observables during experiments whenever reinterpreting measurements in terms of new variables that may be more physically meaningful or analytically convenient.

When applying a transformation to the phase space, it is essential to ensure that the key features of the dynamics, which are represented in the system's phase space geometry, are preserved. These features include, for example, the system's asymptotic behavior, the number of fixed points or attractors, and their homoclinic and heteroclinic orbits. Specifically, it is crucial to verify that the transformation is a diffeomorphism, a smooth, invertible transformation with a smooth inverse within the appropriate domain.

Let $D \subset \mathbb{R}$ and $\phi : \mathbb{D} \to \mathbb{R}$ be a smooth invertible real function onto its image. We now analyze the effect of applying a nonlinear transformation

$$\begin{aligned} \Phi \;&:\; U \to \Phi(U), \quad U \subset D^N, \\ \mathbf{x} \mapsto \Phi(\mathbf{x}) &= \begin{pmatrix} \phi(x_1) \\ \vdots \\ \phi(x_N) \end{pmatrix} = \mathbf{y} \end{aligned}$$

We want to investigate how the system's dynamics are affected and, in particular, check if the manifold-generating property is preserved.

By differentiating with respect to time and applying the chain rule, we can find the equations of motion for the transformed system

$$\dot{\mathbf{y}} = \Phi'(\Phi^{-1}(\mathbf{y})) \sum_{i=1}^{M} c_i(t) W_i(t) \Phi^{-1}(\mathbf{y}) = \sum_{i=1}^{M} c_i(t) \Phi'(\Phi^{-1}(\mathbf{y})) W_i(t) \Phi^{-1}(\mathbf{y}) \tag{11}$$

where $\Phi'(\mathbf{y})$ is a diagonal matrix with diagonal entries $\begin{pmatrix} \phi'(y_1) & \ldots & \phi'(y_N) \end{pmatrix}$ where $\phi'(z_0) = \frac{d\phi}{dz}(z_0)$ and $\Phi^{-1}(\mathbf{y}) = \begin{pmatrix} \phi^{-1}(y_1) & \ldots & \phi^{-1}(y_N) \end{pmatrix}^T$

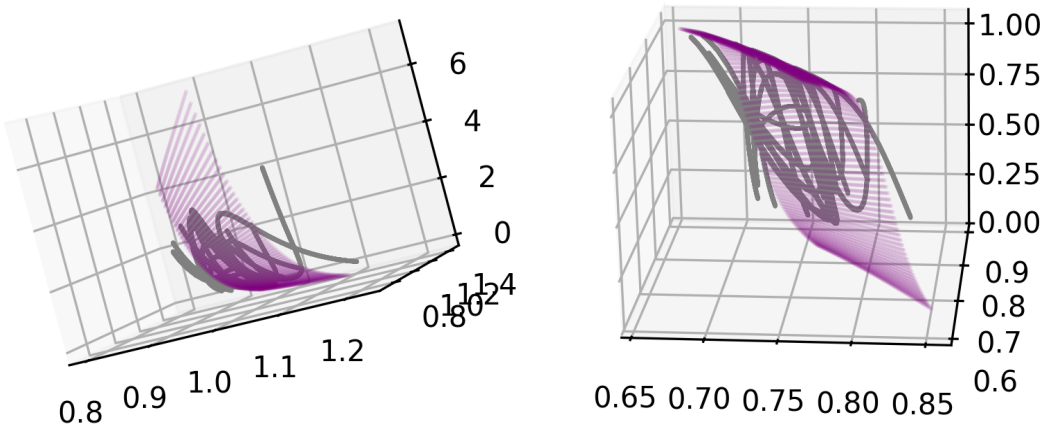


Figure 5: (Left) Trajectories of the original manifold-generating system with two generating vector fields in the three-dimensional phase space starting from the same initial condition and lying on a two-dimensional manifold. (Right) Transformed trajectories and manifold obtained by applying the nonlinear change of coordinates.

Therefore, we obtain as defining vector fields for the system the transformed vector fields under the pushforward of the change of coordinates map Φ, $\Phi_*(W_i\mathbf{x})\Big|_{\phi^{-1}(\mathbf{y})} = \Phi'(\Phi^{-1}(\mathbf{y}))W_i(t)\Phi^{-1}(\mathbf{y})$. We know that a sufficient condition for the transformed system to be manifold-generating is that for all $i, j = 1, \ldots, m$

$$[\Phi_*(W_i\mathbf{x})\Big|_{\phi^{-1}(\mathbf{y})}, \Phi_*(W_j\mathbf{x})\Big|_{\phi^{-1}(\mathbf{y})}] = \mathbf{0} \tag{12}$$

which is verified by the invariance of the Lie brackets under the push-forward and the push-forward being a linear map

$$[\Phi_*(W_i\mathbf{x})\Big|_{\phi^{-1}(\mathbf{y})}, \Phi_*(W_j\mathbf{x})\Big|_{\phi^{-1}(\mathbf{y})}] = \Phi_*[W_i\mathbf{x}, W_j\mathbf{x}]\Big|_{\phi^{-1}(\mathbf{y})} = \mathbf{0} \tag{13}$$

We provide a verification of this fact for this specific case in Appendix B.

To illustrate this construction, we consider a system in $\mathbb{R}^3$ with $m = 2$ commuting linear vector fields, defined by randomly generated commuting matrices W_1 and W_2. The time-dependent coefficients $c_i(t)$ are chosen as sinusoidal functions with different phases. Each trajectory of this dynamical system evolves on a two-dimensional manifold, and the collection of such manifolds foliates the entire phase space.

We then apply a nonlinear transformation Φ defined by applying the scalar function $\phi(z) = \tanh(z)$ to each coordinate. In Fig. 5, we show some trajectories from the original system (Left) and the transformed system (Right) on the corresponding two-dimensional manifolds.

3 Conclusions

We analyzed the properties of linear manifold-generating dynamical systems, a class of systems in which, given an initial condition, each trajectory evolves on an m-dimensional submanifold. These submanifolds together form a smooth foliation of the phase space, structuring the dynamics into m-dimensional leaves.

The equations of motions describing the evolution of single trajectories, given in Equation 5, are expressed as a time-dependent linear combination of m linear vector fields, each modulated by a time-dependent coefficient. These are in fact the vector fields that span the tangent spaces of the integral manifolds of the foliation. This framework is particularly convenient because, by invoking the Frobenius theorem, the existence of the foliation is guaranteed if the m matrices W_i that define the vector fields $W_i\mathbf{x}$ in equations 5 mutually commute, i.e., are simultaneously diagonalizable.

We then explored how this structure can be exploited for data-driven modeling. Given observed trajectories, and assuming an underlying foliation of the phase space, we aimed to recover the governing equations of the evolution of each trajectory in the foliation by optimizing the vector fields and the time-dependent coefficients that describe the evolution along the vector fields. While this kind of per-trajectory fitting could be computationally intensive, we showed that this class of systems admits a closed-form solution in exponential form (see Equation 7). This greatly simplifies the optimization process by skipping the numerical integration of the equations of motion.

By showing that the solutions of a linear manifold-generating dynamical system admit a closed-form expression in exponential form (Equation 7), we highlighted a key computational advantage: the ability to leap directly to any point in time, rather than relying on the incremental stepping characteristic of numerical integration schemes.

We also compared the computational efficiency of using the closed-form solution versus numerical integration for trajectory evaluation during optimization. The results demonstrate that closed-form integration can accelerate computation and might be particularly beneficial for high-dimensional systems.

To extend the expressive power of the model while preserving its core geometric properties, we generalized the system to include nonhomogeneous linear terms, resulting in Equation 9. We found that the the Frobenius conditions still hold, and the manifold-generating property is preserved, if the bias vectors $\mathbf{b}_i$ are chosen such that each $\mathbf{b}_i$ is an eigenvector of each matrix W_j with $j \neq i$ with eigenvalue zero.

Finally, we examined the effect of applying a nonlinear change of coordinates (Equation 11) to the phase space. We showed that, under smooth and invertible transformations applied component-wise, the manifold-generating structure of the system is preserved. This flexibility is particularly useful for adapting to experimental observables or selecting more convenient coordinates, without sacrificing the analytical tractability of the system.

Both such extensions to linear manifold-generating systems do not change the linear nature of the dynamics, but enable writing the solutions for the equations of their trajectories in the simple exponential form that makes them convenient for optimization.

4 Acknowledgements

We are grateful to Janko Petkovic, Laura Bernáez Timón and Maximilian Eggl for organizing the workshop. They brought us all together and fostered an environment characterized by curiosity, exchange and the drive to do science. We equally thank Isabel Cornacchia, Arthur Pellegrino and Angus Chadwick for sharing their knowledge with us, asking the right questions and guiding us in finding the answers. We consider ourselves lucky to have shared this experience with a group of highly motivated participants, everyone of which contributed to the pleasant workshop atmosphere. We further thank the Joachim Herz Foundation for funding the workshop.

References

[1] Vogels, T. P., Rajan, K., & Abbott, L. F. (2005). *Neural Network Dynamics*. Annual Review of Neuroscience, 28(1), 357–376. https://doi.org/10.1146/annurev.neuro.28.061604.135637

[2] Izhikevich, E. M. (2006). *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press. https://doi.org/10.7551/mitpress/2526.001.0001

[3] Fefferman, C., Mitter, S., & Narayanan, H. (2016). *Testing the manifold hypothesis*. Journal of the American Mathematical Society, 29(4), 983–1049. https://doi.org/10.1090/jams/852

[4] Gallego, J. A. (2025). *Neural manifolds: More than the sum of their neurons*. Nature Reviews Neuroscience, 26(6), 312–312. https://doi.org/10.1038/s41583-025-00919-0

[5] Lee, J. M. (2012). *Introduction to Smooth Manifolds* (Vol. 218). Springer New York. https://doi.org/10.1007/978-1-4419-9982-5

Appendix

A Manifold-Generating Nonhomogeneous Linear Systems

We now prove that dynamical systems defined by Eq. 9 where the matrices W_i, $i = 1, \dots, m$ commute and each bias vector $\mathbf{b}_i$ is an eigenvector of each matrix W_j, $i \neq j$, with eigenvalue zero are manifold-gemerating.

Given global coordinates $(x_1, \dots, x_N)$ for the phase space $\mathbb{R}^N$, we denote with ∂_i the i^{th} basis vector $\frac{\partial}{\partial x_i}$ of $T_{\mathbf{x}}\mathbb{R}^N = \mathbb{R}^N$, for $\mathbf{x} \in \mathbb{R}^N$. For the following proofs, we will make use of Einstein summation conventions.

We want to compute the Lie Brackets of pairs of vector fields $f_i(\mathbf{x})^q \partial_q = W_{i,\ p}^{q}\, x^p \partial_q + b_i^q \partial_q$,

$$[W_i\mathbf{x} + \mathbf{b_i}, W_j\mathbf{x} + \mathbf{b_j}]$$

By the bilinearity of the Lie Brackets and the fact that the biases $b_i^q \partial_q$ are constant vector fields, we immediately see that

$$[W_i\mathbf{x} + \mathbf{b}_i, W_j\mathbf{x} + \mathbf{b}_j] = [W_i\mathbf{x}, W_j\mathbf{x}] + [W_i\mathbf{x}, \mathbf{b}_j] + [\mathbf{b}_i, W_j\mathbf{x}]$$

Since the vector fields $W_i\mathbf{x}$ are linear and the biases are constant, we have

- $[W_i\mathbf{x}, W_j\mathbf{x}] = [W_i, W_j]\mathbf{x}$;
- $[\mathbf{b}_i, W_j\mathbf{x}] = b_i^k\ \partial_k(W_{j\ q}^{p} x^q)\ \partial_p = b_i^k W_{j\ k}^{p}\ \partial_p = W_j\mathbf{b}_i$

Therefore,

$$[W_i\mathbf{x} + \mathbf{b}_i, W_j\mathbf{x} + \mathbf{b}_j] = [W_i, W_j]\mathbf{x} - W_i\mathbf{b}_j + W_j\mathbf{b}_i$$

B Nonlinearly Transformed Manifold-Generating Linear Systems

We have the transformed phase space $\mathbb{R}^N$ with transformed global coordinates $(y_1, \dots y_N) = \Phi(x_1, \dots, x_N) = (\phi(x_1), \dots, \phi(x_N))$, with $\phi\colon \mathbb{R} \to \mathbb{R}$ smooth and invertible.

In the general case, if we want to compute the Lie Brackets of two vector fields $F = f^k \frac{\partial}{\partial y_k}$ and $G = g^k \frac{\partial}{\partial y_k}$, we have

$$[F, G](\mathbf{y}) = f^k(\mathbf{y})\ (\partial_k g^l(\mathbf{y}))\ \partial_k - g^l(\mathbf{y})\ (\partial_l f^k(\mathbf{y}))\ \partial_k$$

In this case, $F = \Phi_*(W_a\mathbf{x})\Big|_{\phi^{-1}(\mathbf{y})}$ and $G = \Phi_*(W_b\mathbf{x})\Big|_{\phi^{-1}(\mathbf{y})}$ are the tranformed vector fields through the push-forward of the change of coordinates map Φ of the linear vector fields $W_a\mathbf{x}$, $W_b\mathbf{x}$, with components

$$f^k(\mathbf{y}) = \phi'(\phi^{-1}(y^k))W_a^k{}_l\ \phi^{-1}(y^l)$$
$$g^k(\mathbf{y}) = \phi'(\phi^{-1}(y^k))W_b^k{}_l\ \phi^{-1}(y^l)$$

where $\phi'(z_0) = \frac{d}{dz}\Big|_{z_0}\phi(z)$.

Since, $\frac{\partial}{\partial y_k}\phi^{-1}(y^j) = \frac{1}{\phi'(\phi^{-1}(y^j))}$ we have

$$\begin{aligned}[F, G](\mathbf{y}) = {} & \phi'(\phi^{-1}(y^i))W_a^i{}_k\phi^{-1}(y^k)\left[\frac{\phi'(\phi^{-1}(y^j))}{\phi'(\phi^{-1}(y^j))}W_b^j{}_i\partial_j + \frac{\phi''(\phi^{-1}(y^i))}{\phi'(\phi^{-1}(y^i))}W_b^i{}_l\phi^{-1}(y^l)\partial_i\right] \\ & - \phi'(\phi^{-1}(y^j))W_b^j{}_k\phi^{-1}(y^k)\left[\frac{\phi'(\phi^{-1}(y^i))}{\phi'(\phi^{-1}(y^i))}W_a^i{}_j\partial_i + \frac{\phi''(\phi^{-1}(y^j))}{\phi'(\phi^{-1}(y^j))}W_a^j{}_l\phi^{-1}(y^l)\partial_j\right]\end{aligned}$$

We can relabel the indices and group terms to obtain

$$\begin{aligned}[F, G](\mathbf{y}) = {} & \phi'(\phi^{-1}(y^i))\frac{\phi''(\phi^{-1}(y^j))}{\phi'(\phi^{-1}(y^j))}\left[W_a^i{}_k\phi^{-1}(y^k)W_b^i{}_l\phi^{-1}(y^l) - W_b^i{}_k\phi^{-1}(y^k)W_a^i{}_l\phi^{-1}(y^l)\right]\partial_i \\ & + \phi'(\phi^{-1}(y^i))\Big(W_b^j{}_iW_a^i{}_k - W_a^j{}_iW_b^i{}_k\Big)\phi^{-1}(y^k)\partial_j\end{aligned}$$

We note that, in the expression multiplying ∂_i, we are subtracting the i^{th} row of W_a multiplied element-wise by the i^{th} row of W_b from itself. Hence

$$[F, G](\mathbf{y}) = \phi'(\phi^{-1}(y^i))\Big(W_b^j{}_iW_a^i{}_k - W_a^j{}_iW_b^i{}_k\Big)\phi^{-1}(y^k)\partial_j$$

By hypothesis, we know that $W_a^j{}_iW_b^i{}_k = W_b^j{}_iW_a^i{}_k$, $\forall j, k$, so the result follows.