Inaugural dissertation

for

obtaining the doctoral degree

of the

Combined Faculty of Mathematics, Engineering and Natural Sciences

of the

Ruprecht - Karls - University

Heidelberg

Presented by

M.Sc. Max Ingo Thurm

born in: Bad Muskau

Oral examination: 18.07.2025

Reconstructing Neural Dynamics Underlying Cognitive Flexibility Using Parameter-Evolving RNNs

Referees: Prof. Dr. Daniel Durstewitz
Prof. Dr. Ursula Kummer

Summary

Understanding the dynamic principles that enable the brain to flexibly adapt behavior in changing environments remains a central challenge in neuroscience. In this thesis, I address this question through the lens of dynamical systems reconstruction. I use a reconstruction method, specifically targeted for non-autonomous neural dynamics from multiple single-unit recordings in the rodent medial prefrontal cortex (mPFC) during a probabilistic rule-learning task. To this end, I employ a parameter-evolving piecewise-linear recurrent neural network (pePLRNN), which explicitly incorporates time-dependent changes in the underlying dynamical system (DS). This approach enables the reconstruction of non-autonomous DSs from non-stationary data to characterize of how the neural dynamics evolve across learning.

The approach was first validated on benchmark systems and task-trained RNNs, where it successfully reconstructed the underlying DS. When trained on the hidden state trajectories of RNNs solving artificial rule-learning tasks, the pePLRNN uncovered the dynamic mechanisms by which these networks implemented the learning process.

Applied to electrophysiological recordings from the mPFC of rats, the model successfully reconstructed the non-stationary neural dynamics underlying rule learning. The trained model-generated neural trajectories that exhibited the same decoding properties as the original data. Change points (CP) detected in model-generated trajectories aligned with those observed in the recorded activity. Simulations of neural trajectories under experimental conditions reproduced the behavioral distributions of animals for both rule types.

Analyzing the trained pePLRNN as a functional surrogate model revealed that both rules were implemented via a single stimulus-dependent attracting region that guided neural transients toward the correct decision. During learning, this attracting region, along with the trial-specific parameters and latent neural trajectories, exhibited abrupt changes that preceded the behavioral change point.

This work establishes a principled framework for reconstructing non-autonomous DS directly from empirical data and demonstrates how their analysis as surrogate models can reveal dynamic principles underlying the neural computations supporting cognitive flexibility.

Zusammenfassung

Immer noch ist eine zentrales Problem der Neurowissenschaften diejenigen dynamischen Prinzipien zu verstehen, die es dem Gehirn ermöglichen, Verhalten flexibel an wechselnde Umgebungsbedingungen anzupassen. In dieser Arbeit gehe ich dieser Frage nach, indem ich die nicht-autonome neuronale Dynamiken aus multiplen Einzelzellableitungen aus dem medialen präfrontalen Kortex (mPFC) von Ratten, während sie einer probabilistischen Regel-Lernaufgabe ausführten, rekonstruiere. Zum Einsatz kommt dabei ein parameter-evolvierendes, stückweise lineares rekurrentes neuronales Netzwerk (pePLRNN), das zeitabhängige Veränderungen im zugrunde liegenden dynamischen System (DS) explizit modelliert. Dieser Ansatz ermöglicht die Rekonstruktion nicht-autonomer dynamischer Systeme aus nichtstationären Daten, um zu charakterisieren, wie sich die neuronale Dynamik über das Lernen hinweg entwickelt.

Zunächst wurde der Ansatz an Benchmark-Systemen sowie an rekurrenten neuronalen Netzwerken (RNNs) validiert, die auf künstliche Regel-Lernaufgaben trainiert worden waren. Dabei konnten die zugrundeliegenden dynamische Systems erfolgreich rekonstruiert werden. Beim Training auf die latente Zustandsdynamik dieser RNNs rekonstruierte das pePLRNN die dynamischen Mechanismen, mit denen die Netzwerke den Lernprozess realisierten.

Angewendet auf elektrophysiologische Daten aufgenommen im medialen präfrontalen Kortex von Ratten, rekonstruierte das Modell erfolgreich die nicht-stationären neuronalen Dynamiken, die dem Regel-Lernen zugrunde liegen. Die vom Modell generierten neuronalen Trajektorien wiesen dieselben Dekodierungseigenschaften auf wie die Originaldaten. Zudem traten die im Modell detektierten Strukturbrüche an denselben Zeitpunkten auf wie in den aufgezeichneten neuronalen Aktivitäten. Simulierte Trajektorien unter experimentellen Bedingungen reproduzierten die beobachteten Verhaltensverteilungen der Tiere für beide Regeltypen.

Die Analyse des trainierten pePLRNN als funktionelles Surrogatmodell zeigte, dass beide Regeln durch eine gemeinsame, stimulusabhängige Attraktorregion umgesetzt wurden, die neuronale Transienten in Richtung der korrekten Entscheidung lenkte. Im Verlauf des Lernprozesses veränderten sich diese Attraktorregion, die versuchsspezifischen Parameter sowie die latenten neuronalen Trajektorien abrupt – und zwar bereits vor dem beobachteten Verhaltenswechsel.

Diese Arbeit etabliert ein systematisches Verfahren zur Rekonstruktion nichtautonomer dynamischer Systeme direkt aus empirischen Daten und zeigt, wie deren Analyse als Surrogatmodelle grundlegende dynamische Prinzipien neuronaler Berechnungen offenlegen kann, die kognitive Flexibilität ermöglichen.

Preface

7. Wovon man nicht sprechen kann, muss man schweigen ("Whereof one cannot speak, thereof one must be silent") – with this final proposition, Ludwig Wittgenstein, in the *Tractatus Logico-Philosophicus* [260], attempted to delineate the boundaries of meaningful representational language. He specifically divided the set of possible cases into those that can be expressed using formal language and those that cannot. Later, Alfred Tarski demonstrated in his undefinability theorem that, for any sufficiently expressive formal language, the concept of "truth in this language" cannot be defined within the language itself [236]. This established the necessity of a metalanguage for certain statements that lie beyond the limits of what can be meaningfully expressed. These thought-provoking philosophical contributions open the door to a central question in neuroscience: *Is the answer to the question "How does the human brain work?" within the realm of what can be expressed and understood by the human brain itself?* If so, what would be the nature of such an answer? Although I cannot provide an answer to these questions in this thesis, I aim to illuminate a small segment of the path from our current understanding in neuroscience to the point where we may be able to resolve these questions.

Acknowledgments

First, I would like to express my thankfulness to my doctoral advisor, Prof. Dr. Daniel Durstewitz. I am thankful for the opportunities he created that allowed me to develop a fundamentally new understanding of the processes that underlie thought and learning throughout the course of this work. I am especially grateful for his consistent support, detailed feedback, and the time he committed to guiding me — right up to the very end. Beyond the PhD itself, I want to thank him for the trust and encouragement he offered even before I started my PhD, giving me, as someone with a background in biochemistry, the chance to grow into the field of theoretical neuroscience.

I also thank Prof. Dr. Ursula Kummer, who played a formative role in shaping my scientific path. I am particularly grateful to her for taking on the role of my secondary supervisor, and for the honest, constructive, and forward-looking feedback I received - not only during my PhD, but already during my Bachelor's thesis. Her openness and willingness to listen, have always been a source of reassurance and perspective.

My sincere appreciation goes as well to Dr. Jürgen Pahle and Dr. Georg Köhr for kindly agreeing to serve as examiners for this dissertation. I would also like to thank Prof. Dr. Joachim Haß for joining my Thesis Advisory Committee, along with Prof. Dr. Daniel Durstewitz and Prof. Dr. Ursula Kummer, and for helping to keep this project on track through clear and thoughtful feedback.

I am very grateful to my collaborators for their contributions and insightful discussions. Dr. Florian Bähner provided the electrophysiological data used in this work and was always a motivating presence in our meetings, which continually helped me reconnect with the underlying neurophysiological context. I also wish to thank Prof. Dr. Georgia Koppe for her foundational work on the model formulation, and for the valuable feedback she offered during both my Master's thesis and the early stages of this PhD.

I want to acknowledge the German Research Foundation (DFG) for their financial support within the framework of FOR5159 (Resolving Cognitive Flexibility).

A special word of thanks goes to Prof. Dr. Eleonora Russo, who provided me with my first real entry into theoretical neuroscience. I am so grateful for the time she invested, the intense discussions we had, and the guidance she offered when I fell short, all of which shaped my scientific thinking long before I officially began this PhD. Her continued support during the initial phase of my doctoral studies was equally invaluable.

My deepest thanks go to my highly valued colleagues - people who became friends and companions through the many highs and lows of this journey. They are the ones who shaped the memories that will remain from this PhD. I am especially thankful to Lukas, Flo, and Dr. Manu for the many unforgettable moments that made this experience joyful and deeply enriching, and for their support during difficult times. I also want to thank Janik, Alena, David, and Niclas for the open conversations and shared laughter. Together, they made the office a lively place where ideas could truly take form.

I am especially grateful to those who helped proofread this dissertation: Dr. Manu, Flo, Lukas, and Charlotte. Their support and encouragement in the final stretch gave me the confidence and clarity needed to bring this work to completion.

My heartfelt thanks go to Feruza, who stood by me throughout my PhD. Her support gave me strength, her presence brought calm, and her belief in me kept me moving forward when I struggled.

Finally, I want to express my deepest and most personal gratitude to my family. Without you, this journey would not have been possible. Thank you for your unconditional support, your unwavering belief in me, and for always being there with advice and warm words when I needed them most.

Contributions

The experimental procedures, electrophysiological recordings, and animal work described in the following paragraphs were conducted by Dr. Florian Bähner, with whom I collaborated throughout this PhD thesis. The initial mathematical formulation and the first implementation of the pePLRNN in its first form was developed by Prof. Dr. Georgia Koppe and Prof. Dr. Daniel Durstewitz. For methodological transparency and scientific rigor, I present details regarding animal habituation protocols, training procedures, and experimental implementation and execution (thankfully provided by Dr. Florian Bähner). Additionally, I provide a comprehensive summary of the electrophysiological recording acquisition and preprocessing pipeline used to extract the neural spike trains that build the foundation for this PhD thesis. All subsequent analytical procedures — including further preprocessing, feature extraction, neural state classification, and computational modeling — were performed exclusively by me. This included the entire data analysis pipeline: from raw spike time data transformation over statistical data analysis, and the implementation, validation and tuning of the modeling framework that forms the core contribution of this dissertation. I independently designed and executed all computational experiments and statistical analyses that yielded the results presented in the following chapters of this PhD thesis.

Publication

Durstewitz, D., Koppe, G. & Thurm, M.I. (2023). Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews Neuroscience*, *24*, 693–710.

Contents

Acronyms

BPTT Back Propagation Through Time.

DS Dynamical System.

DSR Dynamical Systems Reconstruction.

DST Dynamical Systems Theory.

GRU Gated Recurrent Unit.

LASSO Least Absolute Shrinkage and Selection Operator.

LSTM Long Short-Term Memory.

ML Machine Learning.

MSE Mean Squared Error.

NMDA N-methyl-D-aspartate.

NN Neural Network.

ODE Ordinary Differential Equation.

PCA Principal Component Analysis.

PFC prefrontal Cortex.

PLRNN Piecewise-Linear Recurrent Neural Network.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

SINDy Sparse Identification of Nonlinear Dynamical Systems.

SR Spatial Rule.

SRES Spatial Rule extinguished site.

SRRS Spatial Rule reinforced site.

VR Visual Rule.

List of Figures

1 Introduction

The central aim of this thesis is to uncover the dynamical principles that enable cognitive flexibility. To this end, the this thesis integrates concepts from multiple disciplines, including systems neuroscience, dynamical systems theory, and machine learning (ML), to establish dynamical systems approach for discovering the governing governing rules underlying the inherent non stationary process behind rule learning. To understanding which neural systems and how they reorganize their activity to adapt behavioral to changing environmental conditions. I begin by introducing the neurobiological and physiological basis, the neural substrate of rule learning, specifically in the medial prefrontal cortex (mPFC) and its role in regulating flexible behavior. This section introduces the relevant anatomical circuits, neuromodulatory systems, and molecular processes that support switching between behaviors. The following section introduces dynamical systems theory (DST) as a formal language for describing neural processes in terms of state space trajectories, vector fields, and attractor structures. ML is then introduced as a methodological toolkit for extracting latent structure from high-dimensional data. Special emphasis is placed on recurrent neural networks (RNNs), which have been used in neuroscience to simulate and interpret dynamic computations in neural circuits. The final section then presents the theoretical underpinnings of dynamical systems reconstruction (DSR) that led to the specific modeling framework employed in this thesis: the parameter-evolving piecewise-linear recurrent neural network (pePLRNN).

1.1 The Neural Basis of Rule Learning - Cognitive Flexibility

One of the most fundamental conditions for survival, for any organism with a nervous system, is the ability to adapt, develop, or switch behavior in accordance with the continuously changing environmental context it is confronted with ([99, 148, 77, 47, 62]). This adaptation of behavioral policies to changes in environmental circumstances, stimulus contingencies, or task rules is called *cognitive flexibility*. Changes in behavioral policy or the acquisition of new rules often occur abruptly rather than gradually, appearing as sudden transitions between distinct behavioral modes ([9, 84, 62]). Cognitive flexibility, as an executive function, is evaluated with behavioral requiring a shift in behavioral policies or response strategies ([96, 74, 70, 156]). In humans, cognitive flexibility is often tested with the Wisconsin Card Sorting Test (WCST), which requires participants to infer and flexibly shift between categorization rules (e.g., color, shape, number) based on feedback ([96, 170, 133]). With lesion studies it was possible to linked the regions of the dorsomedial frontal lobe to substantial impairments during this task, bringing prefrontal circuits and their role in mediating behavioral shifts and error-driven updating into focus ([157, 230, 234]).

Rodent models have provided more insights into the neural basis of cognitive flexibility, particularly within the mPFC ([17, 62, 153, 30, 97, 16, 7]). Especially the prelimbic (PL) subregion has been mainly associated with strategy switching and behavioral adaptation ([175, 184, 95, 62, 204, 7]). In the attention set-shifting task, rats must shift their attention between stimulus dimensions (e.g., from odor

to visual cues) to obtain a reward ([16, 87]). Lesions in the PL selectively impair shifts between stimulus dimensions while maintaining intra-dimensional shifts ([16, 185, 74, 44]).

Reversal learning tasks that specifically require updating stimulus–reward associations, also engage other prefrontal regions like the orbitofrontal cortex (OFC) ([162, 199, 174]). While the OFC is mainly associated with value updating stimulus-outcome pairs ([211]), PL lesions intensify perseverative errors in contexts requiring multiple reversals or hierarchical strategy shifts indicating that PL is critical not only for implementing new rules, but also for maintaining changes in behavioral policies ([186, 72, 163]). Under changing reward contingencies, naive animals flexibly switch between behavioral strategies. However, lesions or inactivation of the PL impairs this flexibility, leading to perseveration on previously leanred strategies ([175, 74]). This type of impairment can be distinguished from that induced by infralimbic (IL) lesions, which decrease the ability to supress previously learned strategies ([186, 7]). Such that PL supports the selecting and stabilizing new behavioral rules, while IL facilitates the suppression of outdated responses and conflict rules ([175, 7]).

The rodent's PL is considered homologous the dorsal anterior cingulate cortex (dACC) in huamns ([15, 100, 242]). Different studies in humans have identified that the dACC is largely involved during strategy shifts and flexible behavior. ([22, 203, 123]). Across different species there is a continuity in behavioral effects when PL functions (or it homologies) are impaired, causing perseverative behavior, a decreased tendency to switching between rules and in general less adaptability to changing enviornmental contingencies ([74, 69, 139, 67, 62, 72, 175, 16, 204]).

Circuit-level mechanisms of flexible rule switching At the circuit level, PL networks show rapid reconfigurations during behavioral transitions, especially when currently executed strategies are not matched by expexted outcome ([72, 74, 62, 10, 122, 130]). Neurons in PL undergo abrupt changes in firing rates during behavioral transition periods, briefly entering a state of high-variability that is associated with exploration of alternative strategies ([62]). Neural recordings have shown that these changes align with behavioral CPs ([62]): as animals switch from a previously reinforced rules, neural variability in PL increases, before stabilizing again when a new rule representation is formed ([62, 122, 130]). This pattern indicates that PL activity could encode uncertainty about the present rule and controls the transition between exploitation and exploration ([130]). Single-unit studies have found that neurons of the PL and the dACC respond specifically to negative feedback, error signals, or conflicting experience, often before behavioral adjustment occurs ([166, 62, 239, 165, 122, 193, 141, 140]). Deep-layer (layer V) pyramidal projection neurons play a specific role in cognitive flexibility. Suppressing these deep-layer (but not superficial) PL neurons disrupts set-shifting, while activating them enhances adaptive rule switching ([225]).

Beyond its internal dynamics, PL interacts with subcortical areas to control rule-dependent behavior. Connections from PL to the striatum, especially the nucleus accumbens (NAc) ([74, 72, 18, 73]), modulate the updating of action-outcome associations. Chemogenetic activation of the PL-NAc pathway improves set-shifting performance by reducing perseverative errors, while inhibiting this pathway impairs

strategy switching ([198, 164, 180]). In additon PL has strong reciprocal connections to the mediodorsal (MD) thalamus which also contributes to flexible rule shifting ([176]). Inactivating the MD thalamus impairs the ability to shift to new rules but has no effect on simple reversal learning, suggesting that this structure supports higher-order context inference ([159]).

At the cellular level PL contains glutamatergic pyramidal neurons (mainly in layers II/III and V) and various types of GABAergic interneurons, including parvalbumin-positive (PV) and somatostatin-expressing cells ([7]). Dopaminergic projections from the ventral tegmental area regulate these circuits through different receptor types: PL pyramidal cells mostly express D1-type receptors ([88]), while D2 receptors appear on both a small number of pyramidal cells and several classes of interneurons ([216, 160, 161, 71]). D1 activation generally increases excitability and promotes rule preservation through stable recurrent activity ([75, 71]), while D2 signaling enhances network flexibility ([212, 71]).

Pharmacological interventions confirm that both receptor types (D1 and D2) are necessary: blocking one of the two dopaminergic receptors (D1 or D2) in PL causes significant impairment in set-shifting, resulting in behavioral perseveration ([71]). In contrast, excessive activation of these receptors shows no improvement in performance, suggesting an optimal dopaminergic level near baseline is optimal for flexible behavior ([71]).

Norepinephrine from the locus coeruleus (LC) also regulates the mPFC through α_1 and α_2 adrenergic receptors on pyramidal cells and interneurons ([173]). Acute stress causes NE levels to rise in the mPFC, and its effects on cognitive flexibility depend on the specific receptor subtype engaged, while α_2 activation enhances network stability and signal-to-noise ratio, excessive α_1-activation induces distractibility and cognitive inflexibility ([173, 151]).

Chronic stress or pharmacological overactivation of NE pathways produces behavioral impairments similar to PL lesions ([151]).

Plasticity in PL circuits are observed during updating internal rule representations and are affected by acute stress ([117, 116]). Blocking N-methyl-D-aspartate (NMDA) receptors in PL prevents the acquisition of new rules but does not impair the execution of already acquired behaviors, indicating that NMDA-dependent plasticity specifically supports updating rather than maintaining rule representations ([226]).

In summary, cognitive flexibility arises from a distributed circuit where PL initiates rule shifts after negative or conflicting sensory feedback, OFC supports value reassignment during reversals, IL stabilizes new policies by inhibiting previously learned strategies, and fronto-striato-thalamic loops coordinate changes in action selection. Behavioral flexibility depends on interactions between excitatory and inhibitory dynamics, neuromodulatory input, and synaptic plasticity in the PL. Dopamine and norepinephrine regulate the PL, by affecting the stability of the current rule representations and allowing a reconfiguration to new rules. This dynamic tuning gives PL the ability to act as a flexible behavioral controller responsible for adaptive behavior. Disrupting PL, or homologous region in humans, consistently impairs adaptive switching across species and tasks, showing its conserved role in regulating cognitive flexibility.

While neurobiological analyses reveal which brain regions and circuits are essential for flexible behavioral control, they do not yet provide a formal language to describe *how* these circuits organize and coordinate their activity over time to implement the cognitive computations necessary for switching behavioral policies. In other words, knowing the anatomical substrate is necessary but not sufficient to explain the computational mechanisms that drive behavioral transitions and rule-learning. Therefor, I next introduce *DST* as a mathematical framework for describing the temporal evolution of complex systems, such as, neural circuits.

1.2 Dynamical Systems

DSs and their mathematical description by DST build a core foundation to understand the physiological and computational processes underlying the complex functions of the brain [114, 108, 109, 195, 258, 60, 246, 229]. DST describes how the state of a system evolves in time and as such offers a rich toolbox to describe, analyze, predict and ultimately understand what natural forces drive a system to transition from one state to another. Many phenomena that can be observed from day to day can be described in terms of the systems dynamics. DST is used to describe system in a variety of different topics: the climate [90], or weather (where the famous Lorenz attractor [136] originates), the flow of traffic (as described by the Lighthill-Whitham-Richards (LWR) model [132]), social sciences [243] and the complex functions of the brain and neurons ([114, 60, 108, 109, 66]).

DS describe the evolution of state variables over time, most generally formulated as ordinary differential equations of the form

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t)),$$

where $\mathbf{x}(t) \in \mathbb{R}^N$ denotes the system's state vector at time t, and $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ defines the system's vector field determining its dynamic evolution ([114, 229]). In discrete time, DS are defined through iterated mappings of the form

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t),$$

where $\mathbf{x}_t \in \mathbb{R}^N$ denotes the state vector at discrete time t, and $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$ is a deterministic function. The evolution of the system is governed by repeated application of this map, such that for an initial condition $\mathbf{x}_0$, the trajectory is given by

$$\mathbf{x}_t = \mathbf{F}^{(t)}(\mathbf{x}_0),$$

where $\mathbf{F}^{(t)}$ denotes the t-fold composition of $\mathbf{F}$ with itself. Formally defined in [181] (adapted from SI of [60]): Let $R \subset \mathbb{R}^N$ be an open subset of $\mathbb{R}^N$, and let the vector field $\mathbf{f} \in C^1(R)$ define the continuous-time dynamics via the differential equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t))$. A DS is defined as a continuously differentiable map $\phi : \mathbb{R} \times R \to R$, called the flow, satisfying the following properties for all $\mathbf{x} \in R$ and all $s, t \in \mathbb{R}$:

1. $\phi_0(\mathbf{x}) = \mathbf{x}$ (identity at zero time),
2. $\phi_{s+t}(\mathbf{x}) = \phi_s(\phi_t(\mathbf{x})) = \phi_t(\phi_s(\mathbf{x}))$ (group property),

3. $\phi_{-t}(\phi_t(\mathbf{x})) = \mathbf{x}$ (invertibility).

While DS are often defined in continuous time, empirical observations like spike times or calcium fluorescence signals are typically sampled at discrete time intervals ([85, 177, 41]), thus, leading to the discrete-time formulation, where the continuous flow is approximated by a time-Δt map. Specifically, given samples $\mathbf{x}_k = \mathbf{x}(k\Delta t)$, the discrete system is defined by a mapping $\mathbf{F}_{\Delta t}$ such that

$$\mathbf{x}_{k+1} = \mathbf{F}_{\Delta t}(\mathbf{x}_k),$$

which approximates the integral of the continuous vector field over the interval $[t_k, t_k + \Delta t]$. The Poincaré map ([182]) reduces the continuous system to a discrete map by recording successive intersections of trajectories with a lower-dimensional cross-section of the state space ([237]). This enables analysis of orbit stability and qualitative dynamics via fixed-points of the induced map.

State Space Representations

The central concept of DST lies in the idea of a *state space*, the space $\mathbb{R}^N$ that contains all dynamical variables required to fully describe the system at any given time t ([114, 54, 259, 155, 60, 229, 246]). Each point $\mathbf{x}(t) \in \mathbb{R}^N$ in state space represents a unique configuration of the system's state, such that its further evolution is completely determined by its current position and the vector field (see Figure 1A for as illustration) ([114, 229, 259]). The *vector field* is a deterministic function $\mathbf{f}$, which defines temporal evolution of states over the whole state space ([114, 54, 259]) (compare with Figure 1D for an illustration). The path of a state in state space is called the *trajectory* (see Figure 1B and C) ([181]). The vector field itself gives rise to dynamical phenomena such as attractors and repellers ([229, 114, 181]).

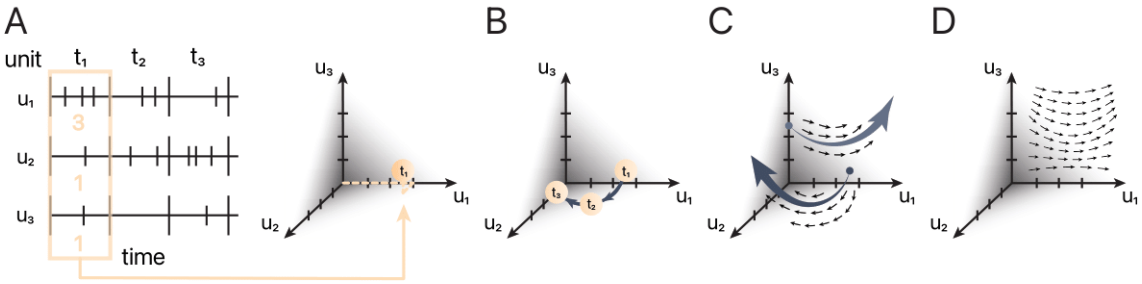

Figure 1: **Illustration of neural state dynamics in state space and time** **A** Three-dimensional neural state space defined by coordinates (u_1, u_2, u_3). Coordinates in state space correspond to the local firing rates. **B** Illustration of a trajectory in state space, as temporal sequence of states. **C** The initial condition together with the vector field define the trajectory. Different initial conditions lead to different trajectories. **(D)** Schematic illustration of the flow field, the geometric structure that governs the temporal evolution of all trajectories.

Attractors, Repellers and Stability

Formally, as defined in [181], an *attractor* $\mathcal{A} \subset \mathbb{R}^N$ of a DS is a closed, invariant set for which there exists an open neighborhood $\mathcal{U} \supset \mathcal{A}$ such that for all $\mathbf{x} \in \mathcal{U}$, the

trajectory $\phi_t(\mathbf{x})$ remains in $\mathcal{U}$ for all $t \geq 0$ and converges to $\mathcal{A}$ as $t \to \infty$, i.e.,

$$\lim_{t\to\infty} \phi_t(\mathbf{x}) \in \mathcal{A}.$$

Moreover, $\mathcal{A}$ qualifies as an attractor if it contains at least one trajectory whose orbit is dense in $\mathcal{A}$. Note that while stable fixed-points, limit cycles, or more complex invariant manifolds can serve as attractors, not all ω-limit sets are attracting. For instance, a saddle point may be the ω-limit set of a few trajectories without attracting a full neighborhood ([229]). Attractors can take various forms, including fixed-points, limit cycles, or more complex sets such as strange attractors (which will be introduced in more depth in later sections) ([114, 229, 54]). Conversely, a *repeller* is an invariant set from which nearby trajectories diverge over time ([114, 229, 54]). These invariant sets divide the state space into regions of convergent or divergent behavior and fundamentally shape the long-term dynamics of the system ([114, 229, 259, 181]).

Cycles and Chaos

Fixed-point attractors are just one specific form of attractors. There are far more complex attractors including closed periodic trajectories known as *limit cycles*. A stable limit cycle is defined as a periodic solution $\gamma(t)$ of the system's dynamics for which all nearby trajectories $x(t)$ satisfy $\lim_{t\to\infty} \mathrm{dist}(x(t), \gamma(t)) = 0$, with $\mathrm{dist}(\cdot,\cdot)$ denoting a suitable metric in state space ([229]). In DS models of single neurons limit cycles are often associated with regular spiking ([114]).

Figure 2a illustrates this phenomenon in a 2D single-neuron model. Here, the trajectory in the (V, R)-state space (representing membrane voltage and a refractory variable) converges toward a stable limit cycle, producing sustained oscillatory dynamics.

At the network level, limit cycle attractors have also been implicated in organizing rhythmic neural activity, for instance in the central pattern generator ([126, 189, 144]). These oscillations emerge intrinsically from recurrent excitation and inhibition ([31]). Neural recordings during slow-wave sleep show low-dimensional oscillatory dynamics, functionally connecting such attractors with memory consolidation during sleep ([31]).

Neural dynamics may also show chaotic behavior, forming *strange attractors*, that attract nearby trajectories. Chaos is widely characterized by its boundedness to a specific region while being sensitive to initial conditions ([76, 227]). More formally, chaos is defined by the exponential diverging trajectories of initially close initial conditions. This phenomenon can be quantified by the *maximum Lyapunov exponent* $\lambda_{\max}$ ([229, 181, 120, 4]). For two initially very close states $x(t)$ and $x(t) + \delta x(t)$, chaos is indicated if

$$\lambda_{\max} = \lim_{t\to\infty} \lim_{\|\delta x(0)\|\to 0} \frac{1}{t} \ln\left(\frac{\|\delta x(t)\|}{\|\delta x(0)\|}\right) > 0.$$

This positive exponent reflects the system's sensitivity to initial conditions, implying that even minimal perturbations can lead to rapid divergence of trajectories.

While $\lambda_{\max} = 0$ characterizes marginally stable limit cycles and $\lambda_{\max} < 0$ corresponds to convergence toward fixed-points, chaotic systems operate at the edge of predictability, with information about past states degrading exponentially over time ([229, 120, 4]).

In transitions between sleeping- and awake sate chaotic dynamics have been observed ([248]). From a computational perspective, chaotic attractors extend the dynamical repertoire of neural systems ([150, 60]). They enable amplification of small inputs, and support separation of internal trajectories in recurrent systems ([149, 150]).

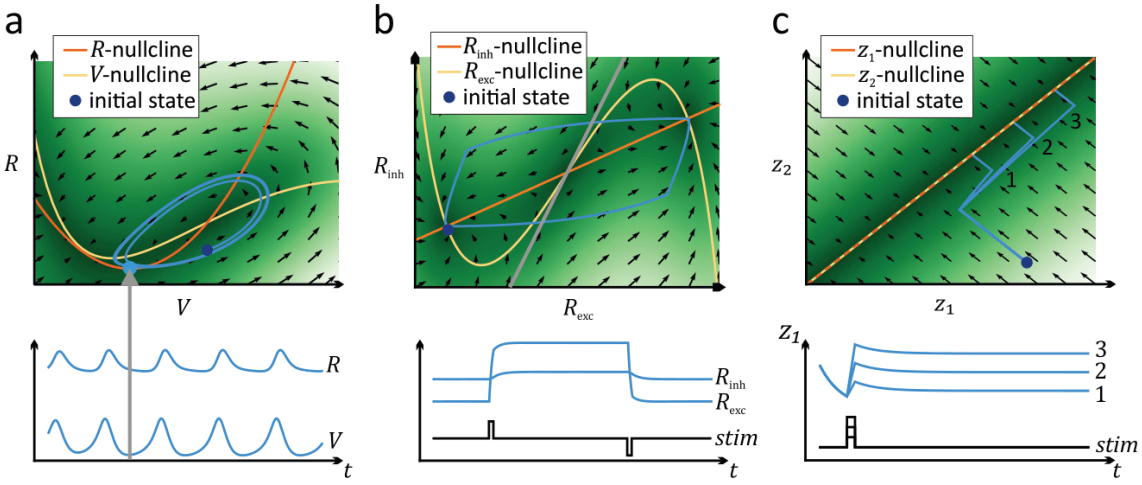

Figure 2: **Illustration of DS phenomena in low-dimensional neural models** **a** The top panel illustrates the state space of a two-dimensional single-neuron model with voltage V and a refractory variable R . The trajectory (blue) converges to a stable limit cycle, reflecting rhythmic spiking behavior. The shading encodes the local magnitude of change. Nullclines (orange and yellow) mark the states where the temporal derivative of either V or R is zero. Their intersection defines an unstable fixed point, from which nearby trajectories diverge and are attracted to the surrounding limit cycle. The bottom panel shows the corresponding time series of the neuron's state variables as the system progresses along the limit cycle. **b** The state space and vector field of a Wilson–Cowan-type neural population model illustrating the bistable dynamics underlying working memory. Each point represents a unique pair of excitatory and inhibitory population firing rates ($R_{\text{exc}}, R_{\text{inh}}$). External inputs drive transitions (blue trajectory) between two stable fixed-points attractors located at the intersections of the nullclines (orange and yellow). Their respective basins of attraction are separated by a boundary (grey line). The central fixed point is unstable, with trajectories diverging from it along the horizontal axis, as indicated by the vector field. **c** A two-dimensional linear neural ODE system producing a line attractor, formed by the exact overlap of its nullclines. Depending on stimulus magnitude, the system evolves from a common initial condition toward different points along this line, enabling memory encoding by converging to a stimulus-dependent states. Adapted from [60]

Bifurcations

Up to this point, we have examined isolated dynamic phenomena of DSs under fixed parameter settings. But what happens when a system's parameters can change? The system's behavior may undergo a qualitative transformation. This type of phenomenon is known as a *bifurcation* [229, 114]. As a system parameter is varied smoothly, the structure of trajectories in state space may shift abruptly: attractors can emerge, disappear, or change their stability [229, 114, 60, 4]. This means that the geometry and topology of the vector field $\mathbf{f}(\mathbf{x})$ that governs the dynamics is altered in a way that fundamentally changes how trajectories evolve over time ([114, 229, 60]).

These transitions are relevant for how neuronal systems work. A well-known example is the shift from resting state to spiking activity in a neuron as input current increases. In this case, a stable fixed point loses its stability and gives rise to a limit cycle, i.e., repetitive spiking ([114, 66, 196, 58]). Which type of bifurcation underlies this change (e.g., saddle-node, Hopf, or homoclinic) determines the neuron's response properties and its functional role in the circuit ([114]). This principle, is not only theoretical, but is used in electrophysiological recordings to characterize cell type by their response ([129]). These differences have been linked to real transitions in cortical neurons under NMDA modulation (see Figure3), where the same cell can exhibit bursting, chaotic firing, or regular spiking depending on the level of NMDA-conductance. [58, 55].

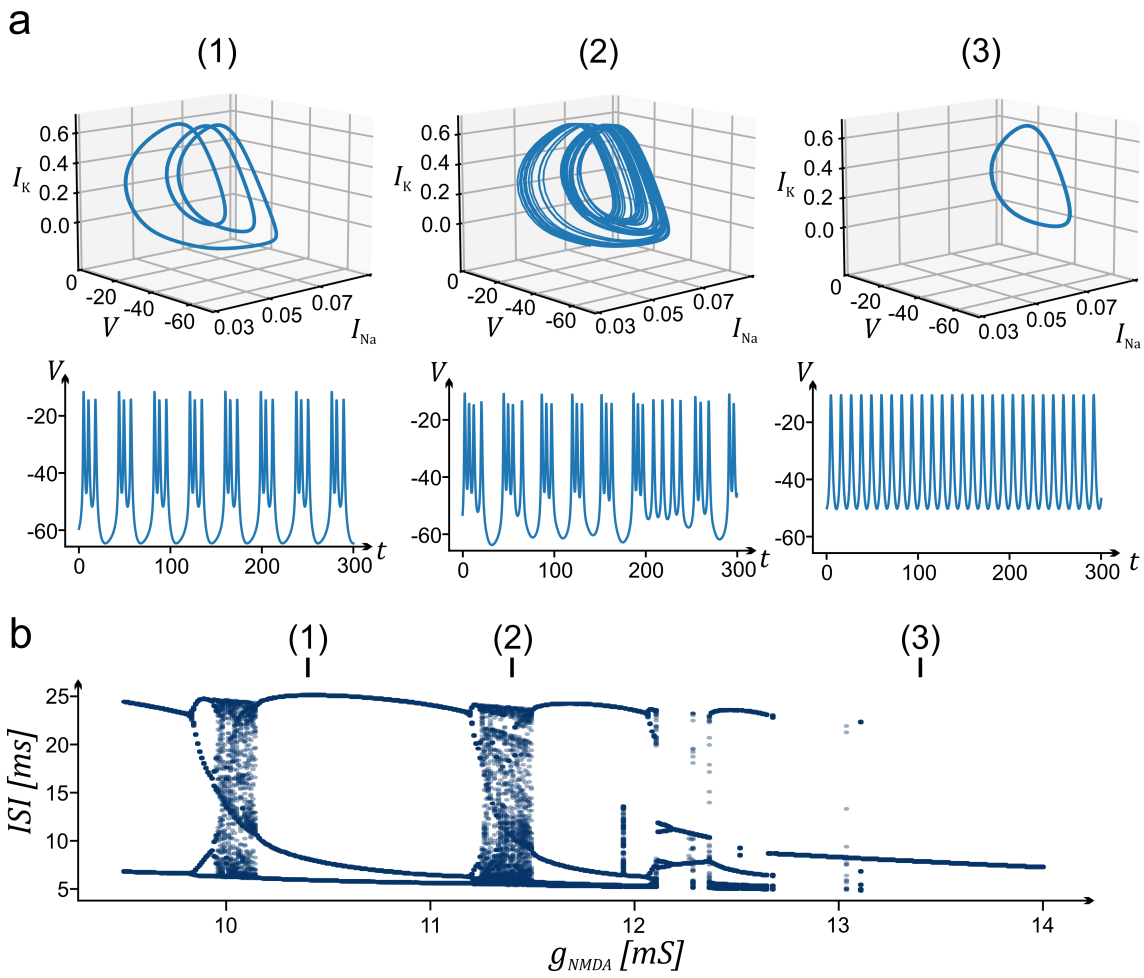

Figure 3: **Dynamical regimes and bifurcations** **a** A three-variable biophysical neuron model [55] with NMDA-modulated input exhibits distinct dynamical regimes depending on NMDA conductance (g_{NMDA}). Increasing g_{NMDA} drives transitions from bursting to chaos to regular spiking, reflecting bifurcations that alter the system's attractor structure. **b** Bifurcation diagram of the biophysical neuron model from a, showing how different NMDA conductance parameter g_{NMDA} influence the inter-spike interval (ISI) distribution. Panels (1)–(3) indicate the corresponding parameter values to the regimes illustrated in (a). (Figure adapted from [59])

Non-Autonomous Dynamical Systems

The DSs and their specific phenomena that have been discussed so far were all *autonomous* meaning that their governing vector field is time-invariant, such that the evolution of the state depends solely on the current state and not explicitly on time. A much more realistic class of DSs in neuroscience are *non-autonomous* DSs in which the vector field depends explicitly on time, i.e.,

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

with $\mathbf{f} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$. The explicit dependence of the vector field on time t, introduces an additional degree of freedom into the evolution of the system ([125],

6]), breaking the time-invariance of autonomous DS. In these systems the initial time t_0 and current time t are required to specify the solution. A simple example (here adapted from [125]) is given by the scalar ODE

$$\dot{x} = -2t\,x,$$

which yields solutions of the form $x(t, t_0, x_0) = x_0\, e^{-(t^2 - t_0^2)}$. Since the solution cannot be expressed purely as a function of $t - t_0$, the system is non-autonomous ([125, 6]).

The flow, in these systems, is described by a two-parameter family of mappings

$$\phi(t, t_0, \mathbf{x}_0) : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N,$$

representing the state at time t resulting from an initial condition $\mathbf{x}_0$ at time t_0. This family satisfies the identity condition

$$\phi(t_0, t_0, \mathbf{x}_0) = \mathbf{x}_0,$$

and a generalized composition rule (also known as the Chapman–Kolmogorov or causality property [[125, 6]]),

$$\phi(t_2, t_0, \mathbf{x}_0) = \phi(t_2, t_1, \phi(t_1, t_0, \mathbf{x}_0)) \quad \text{for all } t_0 \le t_1 \le t_2.$$

This formalism generalizes the one-parameter flow ϕ_t of autonomous systems, and reflects the loss of semigroup structure in $t - t_0$ when time explicitly enters the dynamics ([125, 6]).

To retain some of the structural advantages of autonomous systems, one common strategy is to augment the state space by including time itself as an additional state variable. This results in the extended system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \tau), \quad \dot{\tau} = 1,$$

evolving on the augmented state space $(\mathbf{x}, \tau) \in \mathbb{R}^N \times \mathbb{R}$. The resulting dynamics form an autonomous flow in a higher-dimensional space, which often is referred to as a *skew-product flow*. This enables the application of standard tools from autonomous systems theory, such as the definition of invariant sets or the analysis of stability properties ([125, 6, 156]).

However, the augmentation introduces new challenges: since the time variable τ increases monotonically and without bound, standard asymptotic concepts (like global attractors or invariant measures) require generalization. For example, a time-varying system might possess a moving equilibrium $\mathbf{x}^*(t)$ or a non-stationary periodic orbit, whose shape and location evolve as a function of time. Such time-varying attractors are now represented as geometric objects that remain invariant under the skew-product flow ([125, 6]). These issues are typically addressed using concepts like pullback or forward attractors. A forward attractor A^* is defined as a set that attracts all trajectories as the initial time $t_0 \to -\infty$, holding the current time t fixed. In other words, trajectories initialized far in the past will approach A^* by time t, regardless of the specific starting point within some bounded set ([125]).

This concept is closely related to the pullback concept, which considers a time-indexed family of sets $\{A(t)\} \subset \mathbb{R}^N$ that evolves continuously with t. For each

current time t, the set $A(t)$ attracts all trajectories initialized arbitrarily far back in time. If the system becomes asymptotically periodic, $A(t)$ will inherit that periodicity; if it becomes stationary, the family $A(t)$ converges to a fixed set; otherwise, $A(t)$ continues to deform dynamically with time. Unlike forward attractors, pullback attractors ([89]) do not assume any uniform convergence over future times, making them better suited for describing systems under ongoing or irregular external changes ([125]).

To gain an understanding of the instantaneous configuration of system states at a given time t, forward and pullback attractors do not provide a good description. Therefore, the concept of a *snapshot attractor* was introduced ([200]).

A snapshot attractor characterizes the distribution of states at a fixed time t, obtained by evolving a bounded set of initial conditions $B(t_0)$ from some starting time $t_0 \ll t$. Formally, for a non-autonomous system governed by

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

the snapshot attractor at time t is defined as the image set

$$A_{\text{snap}}(t) = \{\mathbf{x}(t; t_0, \mathbf{x}_0) \mid \mathbf{x}_0 \in B(t_0)\},$$

where $\mathbf{x}(t; t_0, \mathbf{x}_0)$ denotes the trajectory initialized at $\mathbf{x}_0 \in \mathbb{R}^N$ at time t_0, and evolved forward to time t. In the limit $t_0 \to -\infty$, this ensemble converges to a time-dependent distribution that reflects the system's transient structure at time t ([125]).

Snapshot attractors can be seen as temporal slices through a pullback attractor. While pullback attractors track how groups of trajectories settle into sets as time progresses, snapshot attractors capture the geometric configuration of the vector field at a specific moment providing an analysis concept to track changes in local vector fields across time. This distinction between different non-autonomous attractors is particularly relevant for neuroscientific phenomena, where brain dynamics are driven by time-varying internal and external factors (like circadian rhythms, energy homeostasis, or learning and plasticity).

Types of Non-Autonomous Dynamics When considering non-autonomous DSs in neuroscience, it is especially useful to distinguish between different causes of non-autonomy, namely *parameter non-stationarity* (also referred to as parameter drift systems) and *input-driven non-autonomy*. In DSs with input-driven non-autonomy, the system evolves under the influence of external inputs. This case can be modeled by

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

where $\mathbf{s}(t)$ is an exogenous input signal. Here, $\mathbf{s}(t)$ could represent sensory stimuli, task events, or other structured perturbations that directly modulate the system's state trajectory without altering the autonomous update equations themselves.

In contrast, for DSs with *parameter non-stationarity*, the parameters of the system vary as a function of time. This can be commonly written as

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

where the time-dependent parameter vector $\boldsymbol{\theta}(t)$ modulates the underlying vector field. These parameters can typically represent intrinsic system properties. In neuroscience,i.e. these can for instance represent synaptic coupling weights which may change due to external influences or internal adaptive processes like learning, memory formation, or short-term plasticity ([2, 50, 48, 219, 99, 145, 241, 215, 265]).

While both cases involve an explicit dependence on time, they can be interpreted differently in how they modify the system: parameter non-stationarity changes the geometry of the vector field itself, in other words the inner rules, while input-driven non-stationarity modifies the trajectory within a fixed dynamical structure.

In more general cases, both forms of non-stationarity maybe present. These *hybrid systems* combine time-varying parameters and external inputs:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}(t)) + \mathbf{g}(\mathbf{x})\,\mathbf{u}(t),$$

and are particularly relevant in neuroscience, where internal adaptation and external stimulation interact with each other.

These two sources of non-autonomy can each induce transitions into qualitative distinct dynamical regimes, and must be jointly considered when reconstructing or interpreting the underlying system. Fast variability in observed neural trajectories is frequently driven by task-related or other environmental inputs, modeled as external input signals signals $s(t)$ that perturb the latent state $x(t)$ of the system. Such inputs can push the system state across separatrices into new basins of attraction. In perceptual decision-making tasks, for example, a brief or noisy stimulus can displace the trajectory from a neutral fixed point toward a decision-specific attractor, effectively encoding alternative choices ([250, 261, 249]). Or in working memory where once the memory cue (external input) vanishes, the system sustains the information via an attractor state ([61]).

In contrast to external input-driven non-autonomy, neuromodulation, for instance can introduce slow changes to the system's intrinsic parameters $\theta(t)$, thereby reshaping the vector field $\mathbf{f}(\cdot; \theta)$ itself. Neurotransmitters like acetylcholine, dopamine, or serotonin modify cellular excitability ([168]), synaptic connectivity ([105]), or even gene expression ([197]) over minutes to hours shifting the qualitative structure of attractors in the system.

A concrete example is cholinergic modulation in prefrontal cortex. Increased acetylcholine can broaden neuronal tuning curves, reduce spontaneous drift, and stabilize memory-related activity patterns ([19]). Computational models of working memory circuits have shown that enhancing excitability via acetylcholine reduces diffusion of persistent activity bumps resulting in more stable attractors ([61]).

Dopaminergic modulation in frontal-striatal circuits similarly alters the structure of the DS regulating choice behavior. By up-regulating gain or modifying effective connectivity, dopamine can bias the system toward a particular attractor (choice) without presenting any immediate input, reflecting internal state variables like motivation or reward expectation ([167]).

Plasticity as Structural Reconfiguration. While neuromodulation may alter internal parameters episodicaly ([143]), synaptic plasticity like Hebbian learning ([99]), spike-timing-dependent plasticity ([68]) or simply memory formation ([158,

35, 119]) can change the network circuitry over longer timescales. These mechanisms can be interpreted as the core of parameter non-stationarity in neural systems, reshaping the system's dynamics through changes in synaptic weights.

In the motor cortex, learning a new movement sequence can be interpreted as reconfiguration of the attractor structure ([112]). Initially, neural trajectories may be irregular or disorganized across state space. With practice, synaptic changes induce a coherent neural trajectory encoding the motor pattern ([1]).

Developmental processes may as well induce slow parameter drift. Maturation of connectivity, myelination, or aging may all influence the internal parameters.

Internal parameter changes can be also be observed as systematic drifts in firing rates across time. For different plasticity mechanisms are thought to cause a drift in firing rates ([205, 45, 48]). In addition studies have decomposed trial-to-trial variability into components of slowly, varying low-dimensional, parameters rather than attributing all variance to fast noise ([206, 40]).

In sum, DST proofs to be a very versatile tool for understanding neural activity patterns in terms of their temporal organization. It offers a powerful formal language that even under complex conditions like the influence of external stimuli and internal adaptation processes, can provide rich analysis tools that help to identify abstract concepts. Capturing such complexity requires models to account for transient and structural changes in the observed data. Connecting the abstract DST with empirical data motivated the integration of *machine learning* methods, which offer general-purpose tools for constructing data-driven dynamical models. The following section introduces core concepts from ML for modeling neural phenomena.

1.3 Modeling Neural Activity

Machine Learning Principles

Within the following paragraphs I want layout the foundation of ML concepts and the basic terminology that is needed to understand how ML together with DST can be used in neuroscience to understand biological phenomena.

Learning Paradigms ML tasks are usually grouped by data type and model objective. The first category is *supervised learning*, in which a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ of feature–label pairs is provided, and the goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from a hypothesis class $\mathcal{H}$ that minimizes the prediction error on unseen samples ([244]). The learning objective is to minimize the expected risk

$$R(f) = \mathbb{E}_{(x,y)}[L(f(x), y)],$$

where $L(\cdot, \cdot)$ denotes a task-specific loss function ([244]). Since the true distribution is unknown, the expected risk is approximated by the empirical risk

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), y_i),$$

which is minimized during training. For regression, L is typically the squared loss; for classification, cross-entropy loss is usually used (). Generalization is achieved when $\hat{R}(f) \approx R(f)$, meaning the model performs well on unseen data ([94, 54]).

In *unsupervised learning*, only features $\{x_i\}$ are available without explicit labels ([94, 54]). Here the objective is to uncover some form of latent structure in the data or to have a probabilistic model $p_\theta(x)$ that captures the data-generating process ([54, 94]). Therefore most commonly the log-likelihood is optimized:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(x_i),$$

or equivalently the negative log-likelihood is minimized. Unsupervised learning includes objectives like clustering (finding groups in the data), or feature learning via autoencoders, which minimize a reconstruction loss such as $\sum_i \|x_i - g(f(x_i))\|^2$, where f and g are encoder and decoder mappings ([94]). The goal is to model the empirical distribution $P_X(x)$ by learning compressed or constrained latent representations.

A third distinct type of ML is reinforcement learning (RL). Here an agent learns through sequential interaction with an environment to maximize its return([233, 232, 47]). At each time step, it observes a state s, selects an action $a \sim \pi(a|s)$, receives a scalar reward r, and transitions to a new state s'. The goal is to maximize the expected return

$$J(\pi) = \mathbb{E}[R_0], \quad \text{where} \quad R_t = \sum_{\tau=t}^{T} \gamma^{\tau-t} r_\tau,$$

with $\gamma \in (0, 1]$ denoting the discount factor. Unlike supervised settings, there are no direct targets, such that learning is driven by reward signals. The key objective here is to identifying which past actions contributed to observed rewards. Common approaches include policy gradient methods (REINFORCE, [257]) and value-based algorithms ([252]) that propagate reward information backward to optimize the policy ([256]).

Universal Approximation Theory A key result in theoretical ML is the Universal Approximation Theorem (UAT, [135]), which establishes that neural networks (NNs) can approximate arbitrary functions within a wide range of function classes ([43, 111, 37]). Here NNs means a group of artificial neurons in a layer ([146]). Specifically, for any continuous function f defined on a compact subset $K \subset \mathbb{R}^n$, and for any $\varepsilon > 0$, there exists a feedforward NN with a single hidden layer, finite width k, and a non-polynomial activation function $\sigma(\cdot)$ such that the network output $g(x)$ satisfies

$$\sup_{x \in K} |f(x) - g(x)| < \varepsilon,$$

where $g(x) = C\sigma(Ax+b)$, for suitable weight matrices A, C and bias vector b ([111]). This result was independently proven by Cybenko and Hornik in 1989 for sigmoid activation functions([43, 111]), and later generalized to a wider class of nonlinearities ([37]). The main intuition here is that hidden units work as tunable basis functions to approximate the desired function. For instance, sigmoidal activations can approximate step functions, which can then be linearly combined to construct

piecewise-constant or piecewise-linear approximations of any continuous target function ([37]). The rational here is that compositions of affine transformations with nonlinear activation functions densely span the space $C(K)$, and thereby demonstrating the universal approximation quality of a network. Most importantly, the UAT only guarantees the existence of an appropriate set of weights (in the limit case). It does not provide a method for finding these parameters, nor does it specify a minimal width of the network required for an accurate approximation. In practice, this means that sometimes very wide networks have to be used to accurate approximate complex functions([13]). A problem that motivated the use of deep architectures to distribute representational complexity across many hidden layers ([13]).

RNNs, used in the context of sequence modeling and time series analysis, also exhibit this property .

Recurrent Neural Networks RNNs ([202, 64]) form a class of important dynamical models. They are designed to process sequential input by maintaining an internal hidden state that evolves over time ([94]). In their standard discrete-time form, RNNs update a hidden state vector $h_t \in \mathbb{R}^n$ according to the recurrence

$$h_{t+1} = \phi(Wh_t + Us_t + b), \tag{1}$$

where $s_t \in \mathbb{R}^m$ is the external input at time t, $W \in \mathbb{R}^{n\times n}$ the recurrent weight matrix, $U \in \mathbb{R}^{n\times m}$ the input weight matrix, $b \in \mathbb{R}^n$ a bias vector, and $\phi(\cdot)$ a nonlinear activation function applied elementwise (e.g., `tanh`, `ReLU`) ([94, 54]). This nonlinearity enables complex temporal transformations and attractor dynamics beyond linear regimes. An output $x_t = Vh_t + c$ can be defined via an output weight matrix V and bias c, allowing for either sequence-to-sequence or sequence-to-label mappings, depending on whether outputs are taken at each time step or only at the final step.

RNNs can be interpreted as a deep feedforward network unrolled over time, where the hidden state h_t evolves according to a shared nonlinear transition ϕ function. In principle, they can approximate any trajectories of an arbitrary finite-dimensional DS over any finite time horizon ([37, 78, 124]). This was formally proven by Funahashi and Nakamura (1993)[78],who showed that for any continuous-time DS governed by $\dot{x} = F(x)$, there exists a discrete-time RNN of the form $x_{t+1} = f(x_t)$ that can approximate the system's trajectory to arbitrary precision, provided the network has sufficient capacity ([78]). Extending this result, Siegelmann and Sontag proved that RNNs with rational weights and nonlinear activations are computationally universal, capable of simulating a Turing machine ([220]).

RNNs are typically trained using BPTT [253], which unfolds the recurrence over T time steps, resulting in a deep feedforward network with shared weights across layers ([202, 94, 54]). A loss function $L = \sum_{t=1}^{T} L^{(t)}(x_t, x_t^*)$ is minimized with respect to parameters $\{W, U, V, \ldots\}$, with gradients computed by traversing the unrolled computational graph in reverse temporal order ().

Gradients for W accumulate across time:

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \delta_t (h_{t-1})^\top, \tag{2}$$

where $\delta_t = (\epsilon_t + W^\top \delta_{t+1}) \odot \phi'(a_t)$ is the backpropagated error, and $\epsilon_t = \partial L^{(t)}/\partial h_t$ denotes the local output error ([94]).

However, BPTT suffers from vanishing and exploding gradients for large T, as a result of repeated multiplication by the Jacobian $J_t = \text{diag}(\phi'(a_t))W$ during training. If the spectral norm of J_t is less than (or greater than) one, gradients decay (or grow) exponentially over time ([106, 14, 179]). This issue is analogous to instability in deep feedforward networks and is particularly concerning when learning long-range dependencies ([106, 14, 179]).

To address the "vanishing and exploding gradient" problem, gated architectures as the Long Short-Term Memory (LSTM, [107]) extends the vanilla RNN with a memory cell c_t and three gating variables i_t, f_t, o_t that control input, forgetting, and output flow ([107]):

$$i_t = \sigma(U_i s_t + W_i h_{t-1} + b_i), \tag{3}$$
$$f_t = \sigma(U_f s_t + W_f h_{t-1} + b_f), \tag{4}$$
$$o_t = \sigma(U_o s_t + W_o h_{t-1} + b_o), \tag{5}$$
$$\tilde{c}_t = \tanh(U_c s_t + W_c h_{t-1} + b_c), \tag{6}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{7}$$
$$h_t = o_t \odot \tanh(c_t). \tag{8}$$

The additive memory update helps prevent vanishing gradients by preserving information over time when $f_t \approx 1$ and $i_t \approx 0$ ([107]).

The GRU provides a simpler alternative with fewer gates and no separate memory cell. It uses a reset gate r_t and an update gate z_t ([38, 39]):

$$r_t = \sigma(U_r s_t + W_r h_{t-1} + b_r), \tag{9}$$
$$z_t = \sigma(U_z s_t + W_z h_{t-1} + b_z), \tag{10}$$
$$\tilde{h}_t = \tanh(U_h s_t + W_h(r_t \odot h_{t-1}) + b_h), \tag{11}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{12}$$

Both LSTM and GRU architectures enable stable gradient propagation across time and support learning of long-term dependencies, which vanilla RNNs often fail to capture ([106, 107, 39]).

These gated RNN models provided a major advancement toward robust sequence learning architectures in both ML and neuroscience.

Recurrent Neural Networks as Dynamical Systems RNNs create a connection between ML and DST ([60]). When considered with or without external inputs, a vanilla RNN can be described as a discrete-time nonlinear DS. From this perspective, training an RNN corresponds to constructing a DS that can generate the desired trajectories by tuning parameters W, U, and b. After training, all tools from DTS could be in principle used to analyze and interpret the network's internal dynamics ([60]). All fundamental concepts from DST (like fixed-points, attractors, stability, limit cycles, and chaos) apply to RNNs. In particular, nonlinear RNNs can

support multiple fixed-points x^* satisfying $h^* = \phi(Wh^* + b)$. The local stability of such fixed-points is determined by the Jacobian

$$J = \text{diag}(\phi'(Wh^* + b))W, \tag{13}$$

which governs how small perturbations δ_t evolve via the linearized dynamics $\delta_{t+1} \approx J\delta_t$. If the spectral radius of J is less than 1, the fixed point is locally asymptotically stable; otherwise, the point may be repelling or saddle-like ([229, 54]). Nonlinear RNNs can also generate limit cycles, i.e., closed trajectories $\{h_t\}$ satisfying $x_{t+T} = x_t$ for some $T > 0$, which act as attractors for nearby trajectories. In contrast, linear systems only support non-attracting cycles under the condition that all eigenvalues are purely imaginary ($\forall \lambda \in \text{eig}(J) : \Re(\lambda) = 0$).

RNN might be analyzed for a number of properties or dynamic phenomena to insights about their computation:

- **Fixed-point analysis:** Identify equilibrium states and assess their local stability. Stable fixed-points often correspond to memory states or decision attractors ([250, 178]).
- **Attractor structure:** Characterize the geometry of attracting sets like manifolds, cycles, or chaotic regimes that might have a representative function.
- **Limit cycle analysis:** Detect and quantify periodic attractors using tools such as SCYFI ([63]).
- **Controllability and reachability:** Analyze whether input-driven trajectories can access relevant regions of state space.

Understanding RNNs through the lens of DSTs (and vice versa) has become central in computational neuroscience, where DST provides a formal framework to analyze trained models in terms of attractor landscapes or phase space geometry ([60, 231, 11]).

Recurrent Neural Networks in Neuroscience

Early Recurrent Neural Network Models The use of RNNs in theoretical neuroscience was linked to the development of attractor-based memory models. A key contribution came from Hopfield, who proposed a fully recurrent network with symmetric connections that functioned as a content-addressable memory system ([110]). In this framework, the network's activity converges toward stable fixed-points that represent stored memory patterns, enabling reconstruction of complete patterns from partial cues ([110]). This model established a mechanistic link between recurrent dynamics and associative memory, and was later extended to continuous-valued neurons ([251]).

These attractor dynamics were used as models for cortical memory function, opening the possibility for theoretical analyses of storage capacity, stability conditions, and retrieval performance ([5]). Beyond discrete point attractors, models were generalized to support continuous attractors (such as line attractors) which have been proposed to explain graded persistent activity observed in oculomotor

integration tasks ([217]). Such structures enable representation of continuous variables (e.g., spatial location or head direction) via activity patterns constrained to low-dimensional manifolds embedded within the network's state space.

At the same time, early studies also investigated learning mechanisms in RNNs. Williams and Zipser introduced the real-time recurrent learning algorithm, an online method for computing exact gradient updates in continuous-time RNNs ([257]). Applying this framework to model oculomotor control ([217]), they demonstrated that gradient-based adaptation could train a small recurrent circuit to function as an integrator, maintaining stable activity over extended durations. Despite these advances, the practical use of gradient-based training was still limited due to high computational cost and vanishing gradients in long sequences.

As a result, most RNNs used in neuroscience during this period were analytically constructed rather than learned. Ring attractor models for head-direction cells ([221, 190, 223, 208]) and bump attractor models for grid and place coding in spatial navigation circuits highlight this approach ([147, 29, 86]). These models provided mechanistic hypotheses for how stable internal representations could emerge from structured recurrent connectivity, but they did not offer a general framework for task-driven learning.

In the early 2000s, the development of continuous-time RNNs (CTRNNs, [78]) further strengthened the conceptual bridge between DST and neural computation ([224, 131]). These models explicitly formulated RNNs as systems of coupled differential equations, aligning more naturally with the continuous temporal dynamics of biological neurons. CTRNNs were analyzed using tools from nonlinear dynamics demonstrating the diverse behaviors that such networks could produce ([224, 131]). This perspective supported the interpretation of neural computation as the evolution of trajectories in state space, shaped by recurrent feedback.

Architectural and Algorithmic Advances in RNNs After 2000, several key innovations improved the usability of RNNs to model cognitive functions. The development of reservoir computing ([115]) circumvented the instability of training recurrent connections directly ([263]). Two frameworks (Echo State Networks ([115]) and Liquid State Machines ([138])) proposed fixing the recurrent weight matrix with random initialization and training only the output layer ([115, 138]). These models rely on a high-dimensional, nonlinear reservoir to map inputs into state-space trajectories, from which a linear decoder extracts task-relevant signals ([115, 138]).

Within the Liquid State Machine framework, cortical microcircuits were introduced as dynamic reservoirs which can sustain and transform temporal information through intrinsic activity ([137]).

Standard RNN architectures were improved by innovations directly targeting the problems of optimizing these architectures. The introduction of LSTM networks ([107]) addressed the vanishing gradient problem through gating mechanisms that regulate information flow across time ([107]). These mechanisms allow relevant information to be retained across long sequences. Originally developed in ML, LSTMs inspired parallel developments in neuroscience, particularly for modeling working memory and cognitive control. The gating functions in LSTM units have

been likened to biological processes such as basal ganglia-mediated gating of prefrontal representations ([171]) or thalamocortical routing of contextual input ([228]).

RNNs as Computational Models for Cognitive Functions With more powerful architectures and training methods, RNNs have increasingly been used as mechanistic models for cognitive computations ([142, 222, 264, 247, 191, 82]). Rather than assuming specific circuit structures, these models are trained to perform cognitive tasks, and their resulting dynamics are analyzed to identify potential neural mechanisms. An important study demonstrating the effectivity of this approach was conducted by Mante et. al. 2013 [142], in which a RNN was trained on a context-dependent sensory integration task similar to the one performed by primates. The trained network revealed a novel hypothesis: selection and integration of sensory inputs in prefrontal cortex occur through a single dynamical process in which context adjusts "selection vectors" that determine which inputs persist along line attractors and which are canceled through orthogonal relaxation dynamics ([142]). This prediction was later confirmed by neurophysiological recordings from primate PFC circuits ([8]).

Working memory has been a focus for such modeling ([11, 188, 98]). Neural recordings from PFC during memory tasks show persistent activity patterns interpreted as signs of attractor dynamics ([42, 61, 81, 80]). Traditionally, these were modeled using handcrafted circuits with stable fixed-points or line attractors ([154, 79, 81, 61]). RNNs trained on working memory tasks often discover similar solutions through optimization. Binary working memory tasks tend to create discrete attractors ([178]), while continuous memory tasks produce line or ring attractors on low-dimensional manifolds ([113, 218]).

RNN models have also been important in decision-making domains ([250, 3, 91]). Many perceptual and value-based decisions require accumulating noisy evidence over time, a computation naturally supported by recurrent architectures ([249]). Early models showed that mutual inhibition in RNNs can implement winner-take-all dynamics for categorical choice ([250]). Similarly, Carnevale et al. (2015) ([32]) used RNNs to propose a mechanism of how the premotor cortex dynamically adjusts decision-making criteria under temporal uncertainty.

In more recent years RNNs have been used extensively to generate new hypothesis about a variety of neural computations mechanism. Studies focused on timing and the parametric control of neural dynamics. Wang et al. [247] proposed that trained RNNs can implement flexible timing through smooth modulation of internal trajectories. Beiran et al. [12] showed that when inputs are coupled to a low-dimensional contextual signal, networks generalize timing behavior by interpolating between previously learned input regimes. Remington et al. [191] further used RNNs to show that interval and task context adjusted the system's initial condition and input, shaping the geometry of cortical trajectories across trials during a sensorimotor timing.

Beyond single-task training, RNNs have been used to model computational processes of multitasking. RNNs trained on many tasks at the same time developed units with mixed selectivity (similar to prefrontal neuron; ([194]), encoding combinations of task-relevant variables such as rule, context, or evidence. This phenomenon

was shown to be robust across architectures and training regimes ([264]). Dubreuil et al. ([51]) further explored how population structure within RNNs influences computational dynamics during multitasking. They demonstrated that tasks requiring flexible input-output mappings benefit from non-random structures composed of multiple subpopulations. Driscoll et al. ([49]) introduced the concept of dynamical motifs (low-dimensional trajectory patterns) that were selectively recruited across tasks and supported generalization across multiple tasks.

Analysis Tools for Analyzing Trained RNNs from neuroscience The integration of RNNs into neuroscience has been accompanied by the development of tools to analyze their internal dynamics. In theory trained RNNs are fully interpretable by DST, in practice, however they often show high-dimensional, complex behavior that makes their mechanistic understanding difficult ([231]). To address this, DST methods have been adapted to analyze RNN models ([60]; and see above). One approach is fixed point analysis, which aims to identify and characterize the fixed point structure embedded in trained networks ([231, 63]). Sussillo and Barak ([231]) introduced a numerical method to locate fixed-points and linearize the local dynamics around them, allowing the extraction of eigenvalues for stability analysis. This provide important information about the local geometry of the state space and helps characteristic timescales of dynamic modes. While the method presented by Sussilo and Barak relied on numerical computations, certain RNN models, like PLRNNs([53, 127, 128, 24, 104, 63, 152, 101, 23, 26, 210]), can be analytically analyzed for n-cycles with specific algorithms (SCIFY; ([63]). Perturbation analysis offers another perspective on RNN computations. By introducing small perturbations to the internal states and observing the divergence or convergence of the trajectory, one can empirically map the flow field of the system and identify attraction basins ([33]). This technique prompting followup experiments in neuroscience where brief simulations or optogenetic perturbations test the stability and flexibility of circuit dynamics ([172]). RNNs can be analyzed for their Lyapunov spectrum ([65, 245]), to investigate chaotic phenomena.

From Biologically Inspired RNNs to Dynamical System Reconstruction. So far RNNS have used as important tools to generate hypothesis about computational mechanisms, sometimes with specific neurobiological priors ([222]). These modeling efforts share a powerful strategy: train a recurrent model to perform a task, then analyze its internal dynamics as if it were a biological circuit. If the model replicates known neural phenomena, it serves as a plausible mechanistic hypothesis. If it differs, it still offers a testable alternative that can guide empirical research. Despite their success ([60, 11]), these models being trained on task performance, serve primarily to generate hypotheses about the dynamic principles of neural computation, rather than directly providing information about the DST underlying empirical neural data ([60]). This approach marks a conceptual shift: from using RNNs as abstract models of computation, to train them to capture the same geometrical and temporal properties as the DS underlying the observed neural activity. The objective changes from prediction to identification. The next section develops the foundation for DSR.

1.4 Dynamical Systems Reconstruction

At the core of DSR lies the principle that an observed time series contains implicit information about the hidden state variables of the system ([54, 120]). Considering a deterministic DS described by $\dot{\mathbf{z}} = f(\mathbf{z})$ in continuous time or $\mathbf{z}_{t+1} = f(\mathbf{z}_t)$ in discrete time, where $\mathbf{z} \in \mathbb{R}^d$ denotes the full state vector of the system. In neuroscience, this might correspond to membrane voltages, synaptic conductance, or other internal variables. In practice, these internal states are not directly accessible. Instead, one records an output signal $x(t) = h(\mathbf{z}(t))$, representing a (possibly vector-valued) observation of the underlying dynamics ([54, 120]).

Takens' embedding theorem ([235]) provides a foundational result for recovering the system's geometry from such partial observations. Under generic conditions on the measurement function h, a time-delay embedding of a single scalar time series $x(t)$ can reconstruct the topological structure of the original state space ([235, 120]). Specifically, one constructs a delay vector

$$\mathbf{x}(t) = [x(t), x(t+\tau), x(t+2\tau), \ldots, x(t+(m-1)\tau)] \tag{14}$$

using a fixed lag τ and embedding dimension m. Takens (1981) ([235]) proved that, for deterministic systems of dimension d, if $m \geq 2d+1$, the map $\mathbf{z}(t) \mapsto \mathbf{x}(t)$ is an embedding of the original attractor(see Figure4a). That is, it is one-to-one and differentiable, preserving the dynamical degrees of freedom of the system (see Figure4b, for an illustration of)

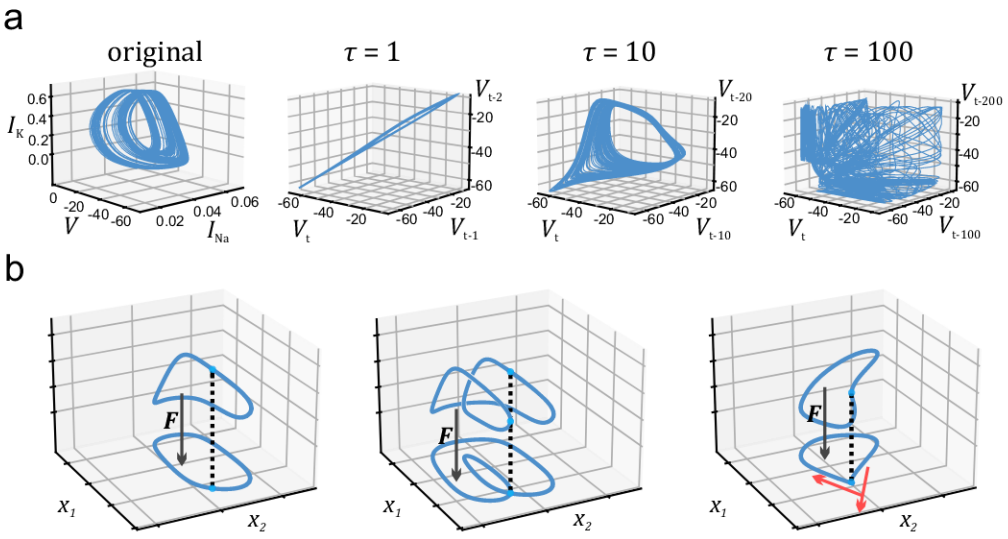

Figure 4: **Illustration of DSR and delay embedding a** Delay embeddings of the original systems trajectory from a single observable (e.g., membrane voltage V_t) into a higher-dimensional space using delayed versions of the signal, such as $(V_t, V_{t-\tau}, V_{t-2\tau})$. Correct choice of delay τ is critical: a too small τ compresses trajectories, too large distorts them. **b** Topological equivalence requires that the embedding map preserves the structure of the original system (left). Violations occur when the mapping is not one-to-one (center) or when its Jacobian becomes singular (right), distorting local geometry. Adapted from [60]

This result ensures that the sequence of observations contains sufficient information to recover the geometry of the underlying system, assuming generic observability ([120]). It builds on Whitney's embedding theorem ([254]) and implies that a topologically faithful reconstruction of the system is possible from scalar observations alone. The key requirement is that the recorded signal must be sufficiently sensitive to all components of the hidden state ([120]).

If certain internal variables never influence the output $x(t)$, then they are unobservable and cannot be recovered ([213, 214]). However, in neural recordings, even single-unit recordings typically reflect a mixture of state variables (also known as mixed-selectivity; [194]), making the system at least weakly observable in many practical settings.

Classical results such as those by Sauer, Yorke, and Casdagli (1991) extended Takens' theory to include multi-dimensional observations and fractal attractors ([209]). These results showed that an embedding dimension greater than twice the correlation dimension D_2 of the attractor suffices to reconstruct its topology. Taken together, these theoretical insights ensure that, under appropriate sampling conditions and observability assumptions, the geometry of a system's state space can in principle be recovered from recorded data ([120, 54]).

However, while delay embedding enables the detection of qualitative features such as fixed-points or periodic orbits, it remains non-parametric ([120]). Therefore it, only provides a geometric reconstruction of the attractor but does not yield an explicit generative model of the underlying system.

Dynamical Systems Reconstruction with Generative Models

To move from purely qualitative and geometrical reconstruction to a generative model capable of reconstructing the underlying DS from data, multiple approaches have been explored ([54, 28, 34, 127, 24, 60, 36, 102]). The main idea behind all these approaches lies in the objective of approximating the the latent DS with a suitable function capable of extracting the governing equations of directly from observed data. One line of work focuses on directly approximating the vector field from the observed time series. *Sparse Identification of Nonlinear Dynamics* (SINDy) [28, 34], address this challenge by approximating the vector field of the system as a sparse linear combination of preselected basis functions (contained in a function library), regularized by LASSO regression ([238]). This leads to a set of analytically tractable differential equations that can provide interpretable insights into the system's mechanistic structure. But only when the time series was actually generated by functions contained in the predefined library. However, the reconstruction performance of SINDy entirely depends on the expressiveness of the chosen function library and breaks down when the true DS cannot be adequately represented by the function library. A more general but less interpretable approach uses NNs to parameterize the vector field. Neural ordinary differential equations (neural ODEs; [36]) model the system as a continuous-time flow, and the evolution of underlying dynamics are inferred via numerical integration. This enables handling of irregular sampling and provides smooth latent trajectories, but at the cost of high computational demands ([92]). They are also not ideal for for reconstructing high-dimensional systems with

complex dynamics, especially because they cannot model functions requiring topological changes in state space ([52]).

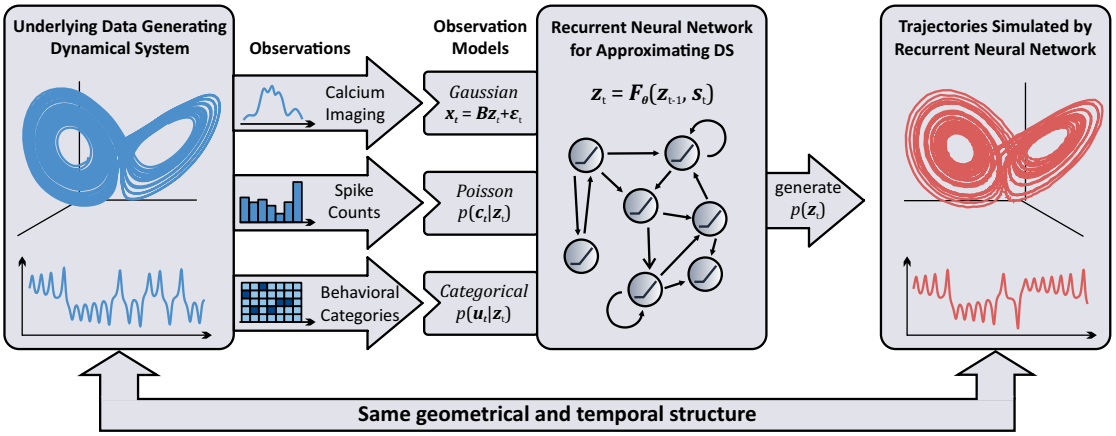

Figure 5: **Conceptual framework for DSR with RNNs observed data** For DSR, RNNs are trained on observed time series from an unknown DS to recover its underlying latent dynamics. Observations like calcium traces, spike counts, or behavioral responses are measurements linked to the latent DS via the specific observation models that define their conditional distributions. Once trained, the RNN can reproduce trajectories with the same geometric and temporal structure as the original system. In DSR the main goal is to have a good approximation of the whole vector field underlying the latent DS (adapted from [60].

RNNs offer a practical alternative to both SINDy ([28, 34]) and neural ODEs ([36]) for reconstructing DSs directly from neural data ([60]). RNNs do not require a predefined function library and can approximate arbitrary nonlinear transition functions through their universal approximation property([37, 78, 124]). In contrast to neural ODEs, they operate in discrete time and do not rely on numerical integration, avoiding high computational costs and numerical instability. Their state evolution directly captures sequential dependencies, making them well-suited for modeling DSR. In contrast to the RNNs from the previous section**??**, RNNs for DSR are not trained on an artificial cognitive task but directly on data in the form of uni-modal ([53, 127, 24]) or multi-modal data ([128, 27, 25]), with specialized encoder models ([128, 24]). After training, the RNN can generate data that exhibits the same temporal and geometrical properties like underlying data-generating system (see Figure**??**). However, applying vanilla RNNs to DSR introduces challenges. Standard training algorithms such as BPTT ([253]) suffer from the exploding-vanishing gradient problem ([106]). To control these limitations, specialized trained algorithms have been developed ([152, 24, 104]). Approaches such as teacher forcing, where observed data periodically replace model-generated states during training (Sparse teacher forcing; [24, 152]), help to stabilize the latent dynamics and prevent divergence ([152]). An alternative to sparse teacher forcing is generalized teacher forcing ([104]), where model-generated and observed states are continuously interpolated throughout training. This approach is particularly effective as the interpolation factor can be dynamically adapted over time to control for exploding gradients

during training ([104]). Additionally, explicit regularization terms have been introduced that penalize deviation from expected long-term behaviors, for instance the manifold-attractor regularization [210]. These improvements in training RNNs on noisy, or chaotic data. Form the basis for the pePLRNN framework further developed in this work.

Model Validation in DSR

Reconstructing a DS is not about fitting the observed time series data with a low prediction error ([127, 60, 24, 27]). In case of chaotic systems the prediction error is inherently meaningless, due to the sensitivity property of the chaotic system ([127, 262]) and can even lead to misleading conclusions about the quality of the reconstruction (see Figure6A). Therefore, multiple criteria are used to validate DSR model ([127, 60]) on empirical data. A better suited test is whether invariant measures and the qualitative structures of the true system can be reproduced by the DSR model ([235, 60, 27]). An invariant measure introduced by Koppe et al. (2019)([127]) is based on the Kullback–Leibler divergence (refered to as the *state space distance*) between the spatial distribution of model-generated states and the distribution of true states ([127, 24, 27, 60]). A good reconstruction has a small KL divergence (see Figure6B), indicating the model spans the same regions in state-space with similar frequency ([127, 24, 27, 60]). A power spectrum comparisons can be used for validating the temporal properties of the reconstructed system (for instance the Hellinger or Wasserstein distance; [60]). This ensures the model has captured the correct temporal componets of the true system. The qualitative structure can be assessed by comparing topological features like the number and type of attractors ([24, 63]). A concept sometimes invoked here is topological conjugacy ([235, 120]). Two systems are considered equivalent if there exists a continuous invertible change of coordinates mapping one system's trajectories to the other's (see Figure6; [209, 120]).

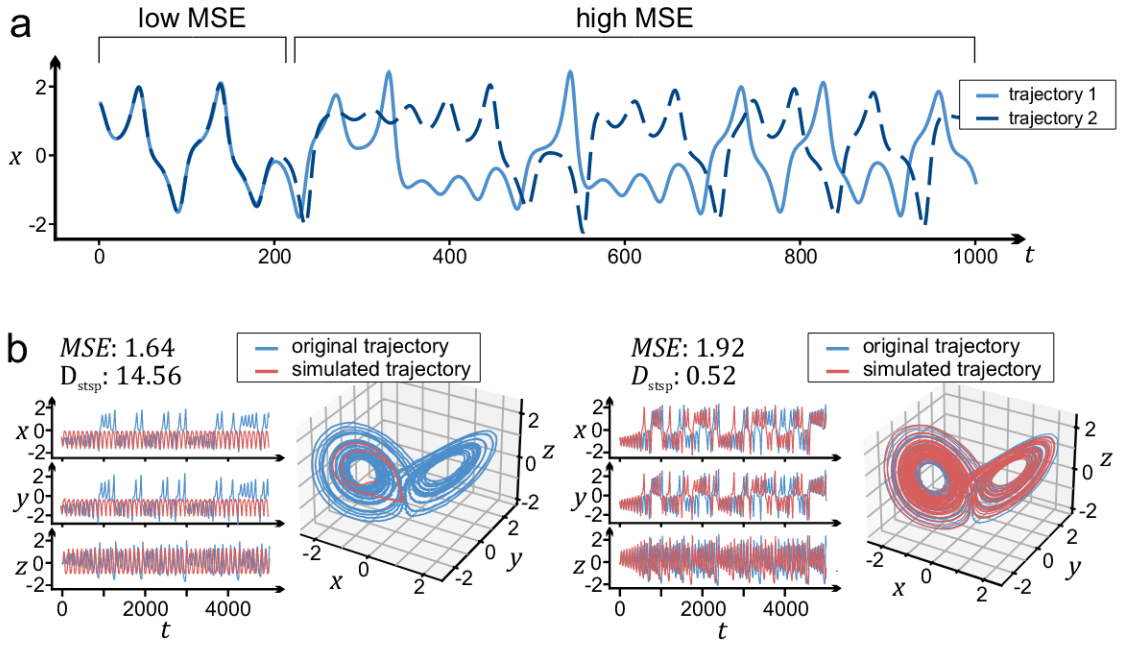

Figure 6: **Reconstruction measures for chaotic system** **a** The Mean-squared error (MSE) is inadequate for evaluating DS reconstruction in chaotic systems, as small initial differences lead to diverging trajectories despite underlying dynamical similarity. **b** Geometric measures such as state-space distance (D_{stsp}) better capture reconstruction quality by comparing the overall structure of true and generated trajectories. Low MSE may mislead if the reconstructed system mimics only superficial features (e.g., frequency), while high MSE may still arise despite correct dynamic geometry. Adapted from [60]

Practical Challenges

Reconstructing the neural dynamics from neuro-physiological recordings poses a far greater challenge than well-behaved systems such as the Lorenz attractor ([136]). Neural systems present several complicating factors that violate key assumptions (i.e. time invariance, noise free and fully observed; [120]) of classical DSR ([60, 120]) and require methodological adaptation.

1. **Partial Observability**
   In neuroscience, datasets rarely contain all relevant state variables. For instance, electrophysiological recordings may capture the activity of only a small subset of neurons in a larger circuit, while functional Magnetic Resonance Imaging (fMRI) signals represent spatially averaged activity from millions of neurons. This results in a many-to-one projection from the true state space to the observable signal.

2. **Noise and Stochasticity**
   Neural signals are inherently noisy, with variability arising from thermal fluctuations, probabilistic synaptic transmission, instrumentation noise, and unobserved modulatory inputs [46]. This noise can obscure the structure of the

underlying attractor, inflate dimensionality estimates, and invalidate deterministic reconstruction theorems.

3. **Non-Stationarity**
   Parameters governing the neural dynamics drift over time due to learning, changes in physiological condition, or external influences. This violates the assumption of stationarity, i.e., that the system is governed by a time-invariant function f.

1.5 Aim of this Thesis

The current gap across all related fields is that no existing model is capable of reconstructing the non-autonomous DS underlying non-stationary neurophysiological data while incorporating both conceptual forms of non-stationarity, namely, external sensory inputs and internal parameter reconfigurations. This means there is no model available that accurately takes the natural conditions of learning into account. Moreover, there is a lack of specific validation criteria for assessing the reconstruction quality of a non-autonomous DSR model trained on neurophysiological data. To date, no effective framework exists for analyzing such models to gain mechanistic insights into the DS principles underlying rule learning in the rodent brain. Filling these gaps is the central aim of this thesis, to bridge the gap between experimental neuroscience and DSR by presenting a mechanism in the language of DST for rule learning in the rat's mPFC, reconstructed from highly noisy, non-stationary multiple single-unit recordings. To achieve this, I employ the parameter-evolving piecewise-linear recurrent neural network (pePLRNN) to reconstruct non-stationary time series data as a piecewise-stationary system. This is done by introducing *snapshot parameters* (or time-dependent parameters) to the existing autonomous formulations of PLRNNs [53, 127, 210, 128, 24, 104, 25, 23, 27], which, together with the consideration of external inputs, allow for capturing the two most abundant sources of non-stationarity in neural data. I demonstrate that the pePLRNN can serve as a functional surrogate for non-stationary ground truth data and the recorded MSU data by proposing a set of validation criteria that account explicitly for the non-stationary in the observed data. Using this surrogate framework, I show that rule-learning in all animals follows a common mechanism: a single stimulus-dependent attracting region that guides the neural trajectory during task performance toward the correct decision. I further analyze how these attracting regions evolve over the course of learning. Finally, I demonstrate that even small changes in task design can significantly affect the dynamic mechanism used to solve the task and that modeling assumptions of external input can influence the reconstruction outcome. Taken together, this thesis aims to provide a general advancement in the understanding of neural systems as non-autonomous DSs.

2 Methods

2.1 Behavioral Task and Neural Recording

Animals

For the experimental protocol, six male adult Sprague-Dawley rats (Charles River, Sulzfeld, Germany) were used that were 8 weeks of age upon acquisition. The rule switching task began when animals reached 4-6 months of age. Initially, animals were housed in standard macrolon cages (55 × 33 × 20 cm) with group housing (4 animals per cage). Following silicon probe implantation, animals were individually housed in identical cages with custom-designed protective lids to prevent implant damage or displacement. To maintain consistent motivational states during experimental sessions, Dr. Florian Bähner established a controlled feeding regimen (20 g per animal daily, provided after the experimental sessions) that allowed animals to develop a normal weight gain while ensuring sufficient motivation during experiments. Water was provided ad libitum throughout the experimental timeline. The housing facility maintained a controlled 12-hour light/dark cycle (07:30-19:30 light phase), with all experimental procedures conducted exclusively during the light phase. All experimental protocols adhered to national and international ethical standards for animal research, were conducted in compliance with the German Animal Welfare Act, and received prior approval from the appropriate regulatory authority (Regierungspräsidium Karlsruhe, Germany; approval number G4-16).

Behavioral Training Protocol

Rats were trained systematically to perform the probabilistic rule switching task. Initially, rats were trained to press levers for reward in a standard operant chamber (21 × 29 × 24 cm), while the main task was conducted in a larger custom-made chamber (30 × 48 × 41 cm). The experimental apparatus featured two retractable levers positioned on either side of a central food delivery tray, with cue lights mounted above each lever and a house light in the upper corner. All chambers provided light and sound isolation, with constant background ventilation noise to minimize external distractions.

The task implementation was controlled through MedPC-IV software with custom MedStat Notation code (MedAssociates Inc.). During initial training phases, correct responses were rewarded with 80 μl of sweetened condensed milk (Milchmädchen, Nestlé), while the main task utilized 45 mg food pellets (BioServ) as rewards.

The rule-switching paradigm was adapted from [74], with a modification: rather than deterministic reward feedback, probabilistic reinforcement was implemented (80% reward probability for correct responses, 20% for incorrect responses) to increase task complexity and better approximate real-world decision contexts.

Rats were initially trained to respond equivalently to both levers when presented individually prior to the task. Animals first learned a visual discrimination rule (VR), in which reward delivery was conditioned on pressing the lever located beneath an illuminated cue light. Training continued until animals reached a predefined performance criterion of at least 80% correct responses.

On the day following VR acquisition, animals began each session with the VR as a baseline performance block until reaching the performance criterion of 80% correct responses after performing at least 30 trials. A rule switch, without any explicit cue to the animal, was implemented, requiring animals to learn a Spatial Rule (SR). Under the SR, rewards were delivered for pressing a specific lever side (left or right), independent of cue light location - the *cue-to-place* shift.

A rule was considered to be learned when an animal reached a performance threshold of at least 80% correct responses within the last 20 trials. Each rule condition was maintained for a minimum of 20 and a maximum of 250 trials. After reaching the performance criterion, the rule was switched again on the following day, starting with a SR baseline block followed by the visual rule (VR), thereby forming a *place-to-cue* shift. Each animal completed a total of six such rule transitions.

Individual trials followed a fixed temporal structure. At trial onset, a single cue light was randomly activated on either the left or right side of the chamber. After a delay of 3 seconds, both levers were extended into the chamber. Animals had a 10-second response window in which to press one of the levers. The cue light remained on for the whole delay and response period until the animals made a choice or the response period was over. Correct responses were probabilistically rewarded with an 80% chance of pellet delivery, whereas incorrect responses yielded a 20% chance of reward. Trials with no response within the response window were scored as omissions. Following each trial, levers were retracted and a fixed inter-trial interval of 20 seconds was imposed before the onset of the following trial.

Surgery

Six animals underwent surgical implantation of microelectrodes after they had acquired the ability to respond equivalently to the individual presentation of both levers. The 64-channel silicon probes (chronic P1-probe; 4 shanks, 16 channels per shank; Cambridge NeuroTech, Cambridge, UK), mounted on a nano-Drive microdrive system (Cambridge NeuroTech), were implanted into the right prelimbic region of the mPFC. The center of the probe array was positioned at stereotaxic coordinates anterior-posterior (AP) +3.0 mm, mediolateral (ML) +0.6 mm, and dorsoventral (DV) −3.0 mm from the cortical surface. Surgeries were performed under isoflurane anesthesia (2.0–2.5%). A bone screw placed above the cerebellum served as ground reference.

Electrodes were moved only when signal quality deteriorated. Electrode placement within the prelimbic cortex was verified histologically. Animals were deeply anesthetized and transcardially perfused with 4% buffered formalin. The entire head, with electrodes in it, was stored in formalin for three weeks to preserve the electrode tracks. Brains were extracted and sectioned using a vibratome. This procedure enabled visualization of electrode tracks without the need for additional staining.

Electrophysiology

Animals were allowed to recover for a minimum of seven days before being habituated to the recording setup and reintroduced to the behavioral environment through

additional training sessions. The set-shifting paradigm started no earlier than 14 days following electrode implantation.

Neuronal activity was recorded simultaneously from multiple single units using a 64-channel RHD2164 amplifier connected to an RHD2000 USB interface board (Intan Technologies LLC, CA, USA). Channels were digitized at 16-bit resolution, sampled at 30 kHz, and band-pass filtered between 0.1 Hz and 7500 Hz. Behavioral event time stamps—including cue light onset, lever presentation, and lever presses—were transmitted from the Med Associates behavioral control system to the Intan acquisition system, allowing precise alignment of behavioral events with neural recordings.

Raw Data Acquisition and Preprocessing

Raw electrophysiological data were preprocessed prior to spike sorting. Signals were band-pass filtered between 600 and 6000 Hz using a Butterworth filter (implemented via the `filtfilt` function in MATLAB). To suppress global noise and remove shared artifacts, the median signal across all channels was subtracted at each time point.

Spike detection and automatic sorting were performed using the Klusta software suite (`https://github.com/kwikteam/klusta`), followed by manual curation in Klustaviewa (`https://github.com/klusta-team/klustaviewa`; Rossant et al., 2016 [201]). During manual curation, putative units detected by individual templates were inspected and discarded if classified as noise based on non-physiological waveform shapes or pattern of activity across channels.

Units with low-amplitude spikes, waveform heterogeneity, or evidence of refractory period violations were labeled as multi-unit activity and excluded from further analysis. To identify potential redundancies, each unit was compared to spatially adjacent clusters and merged when justified by waveform similarity, spike train correlations, or drift patterns. Additionally, units were excluded if more than 1% of interspike intervals (ISIs) were shorter than 2 ms. Units passing all criteria were considered to represent the spiking activity of single neurons.

Spike convolution and unit filtering

The following preprocessing of spike data and unit selection was implemented and conducted by myself. Raw spike times were transformed into continuous firing rate estimates to be learned by the DSR model. Spike times were organized into a matrix $\mathbf{S} \in \mathbb{R}^{N \times T}$, where each row corresponds to a unit i and each entry marks a spike time in milliseconds. For each unit, inter-spike intervals were computed and used to estimate a unit-specific kernel width. The standard deviation of these intervals was scaled by a fixed factor $\sigma_{\text{scale}} = 1$ to obtain

$$\sigma_i = \sigma_{\text{scale}} \cdot \text{std}(\Delta \mathbf{s}_i),$$

where $\Delta \mathbf{s}_i$ denotes the inter-spike intervals for unit i. Each spike was convolved with a Gaussian kernel to generate a time-continuous firing rate:

$$K_i(t_b, t) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(t_b - t)^2}{2\sigma_i^2}\right),$$

$$r_i(t_b) = \sum_{t \in \mathcal{S}_i} K_i(t_b, t),$$

with $\mathcal{S}_i$ denoting the spike times of unit i. Each unit was separately convolved and standardized afterwards.

Units with low firing rates for the entire session were excluded from further analysis. Units were excluded if their single-trial-averaged firing rate was below 1 Hz in 30% of all trials. This filtering procedure ensured that only active units contributed to the model reconstruction.

Stable Performance Periods

Stable performance periods for each rule were defined as the final 20 trials before the rule change, during which behavioral accuracy was at least 80%, and the last 20 trials of the recording session. In two sessions, the second performance period was shifted to earlier trials because of the animal's disengagement from the task. For dataset 10, the stable period was shifted by 25 trials, and for dataset 17 by 23 trials, ensuring that only trials with task engagement were included.

Behavioral Performance Evaluation for Cue-Choice Association

To determine whether the cue stimulus influenced the choice of the animal during the two rule conditions, I used a chi-squared test of independence on the behavioral choices and the cue stimulus. This test was applied to behaviorally stable periods (see Methods 2.1) of each session. Omission trials were excluded from the analysis.

The test evaluates whether the distribution of behavioral choices is independent of the cue stimulus location. Let the set of binary stimulus values be denoted by $S \in \{0, 1\}$ and the corresponding binary choice values by $C \in \{0, 1\}$, for a given rule condition. A 2×2 contingency table is constructed from the joint counts of stimulus–choice pairings:

$$\begin{bmatrix} n_{00} & n_{01} \\ n_{10} & n_{11} \end{bmatrix},$$

where n_{ij} represents the number of trials with stimulus $S = i$ and choice $C = j$. Under the null hypothesis cue and choice are independent, in which case the expected frequencies E_{ij} for each cell are computed as

$$E_{ij} = \frac{n_{i\cdot} \cdot n_{\cdot j}}{N},$$

where $n_{i\cdot}$ and $n_{\cdot j}$ are the marginal sums and N is the total number of trials.

The chi-squared statistic is then given by

$$\chi^2 = \sum_{i=0}^{1} \sum_{j=0}^{1} \frac{(n_{ij} - E_{ij})^2}{E_{ij}},$$

with one degree of freedom. Statistical significance was assessed using a two-tailed test at $\alpha = 0.05$. If the null hypothesis of independence was rejected ($p < \alpha$), the behavior was classified as VR behavior, otherwise (in case of significance), the behavior was labeled as SR behavior. The test was conducted using the `scipy.stats.chi2_contingency` function from the `SciPy` library.

Dataset Inclusions

In total I obtained 24 sessions for analysis. Two of the 24 sessions were excluded: one due to a technical error, another because the animal disengaged from the task for an extended period. On average, a session lasted 1 h 29 min $\pm$ 42 min and comprised 193 ± 95 trials. In each session, 41 ± 15 units were recorded.

2.2 Computational Modeling Framework

Dynamical Latent Space Model

To reconstruct DS where the underlying system's parameters cannot be assumed to be stationary, I used a modified version of the clipped shallow piecewise-linear recurrent neural network (clshPLRNN) from [104]. This model, called the parameter-evolving PLRNN (pePLRNN) reconstructs latent dynamics directly from non-stationary time series data $X \in \mathbb{R}^{T \times N}$ by introducing time-varying connectivity matrices, denoted as $\mathbf{W}_1^{(k)}$ and $\mathbf{W}_2^{(k)}$ linked to a specific temporal segment X^k, called a trials, of larger time series, the session. The system dynamics are governed by the equations:

$$\mathbf{z}_{t+1} = \mathbf{A}\,\mathbf{z}_t + \mathbf{W}_1^{(k)}\Big[\phi\Big(\mathbf{W}_2^{(k)}\,\mathbf{z}_t + \mathbf{h}_2\Big) - \phi\Big(\mathbf{W}_2^{(k)}\,\mathbf{z}_t\Big)\Big] + \mathbf{h}_1 + \mathbf{C}\,\mathbf{s}_t, \tag{15}$$

$$\mathbf{x}_t = \mathbf{I}\,\mathbf{z}_t, \tag{16}$$

where $\mathbf{z}_t \in \mathbb{R}^M$ represents the latent states at time t, and $\mathbf{A} \in \mathbb{R}^{M \times M}$ is a diagonal matrix encoding their intrinsic time constants. The trial-dependent connectivity parameters, $\mathbf{W}_1^{(k)} \in \mathbb{R}^{M \times L}$ and $\mathbf{W}_2^{(k)} \in \mathbb{R}^{L \times M}$, facilitate the modeling of non-stationary dynamics as changes in the flow field. Here, $\phi(\cdot)$ corresponds to the ReLU activation function, while $\mathbf{h}_1 \in \mathbb{R}^M$ and $\mathbf{h}_2 \in \mathbb{R}^L$ serve as bias terms. External inputs, $\mathbf{s}_t \in \mathbb{R}^U$, are integrated into the model via the linear transformation $\mathbf{C} \in \mathbb{R}^{M \times U}$. The observable data $\mathbf{x}_t \in \mathbb{R}^N$ are obtained through an identity mapping of the latent states.

Overall, the pePLRNN is characterized by the parameter set

$$\Theta = \{\mathbf{A}, \{\mathbf{W}_1^{(1)}, \ldots, \mathbf{W}_1^{(K)}\}, \{\mathbf{W}_2^{(1)}, \ldots, \mathbf{W}_2^{(K)}\}, \mathbf{h}_1, \mathbf{h}_2, \mathbf{C}\}.$$

Model Training

The DS model described by equations 15 and 16 is trained using a training protocol based on Generalized Teacher Forcing (GTF), sub-sequence sampling, and an annealing schedule as detailed in [104]. GTF is an advanced training protocol to allow for a controlled gradient propagation during training. Essentially it interpolate the model prediction with a teacher-forcing signal from the data. This practically controls the cumulative product of Jacobians during model training effectively preventing exploding gradient [104]. The exact training protocol with all hyperparameters can be found in the section5. Let $\mathbf{x}_{1:T}$ denote the observed time series segmented into K trials. The training procedure is composed of the following steps:

Annealing Schedule During training, GTF is applied by interpolating between the predicted latent states $\mathbf{z}_t$ and the data-derived ground truth states $\mathbf{z}_t^{\text{true}}$:

$$\tilde{\mathbf{z}}_t = (1-\alpha)\,\mathbf{z}_t + \alpha\,\mathbf{z}_t^{\text{true}},$$

with the annealing parameter α initialized at 0.5. It is then exponentially decayed to 0.1 during the first 10% of training epochs and further to 0.001 over the remaining epochs.

Sub-sequence Sampling. At each epoch, sub-sequences of length $\tilde{T}$ are randomly sampled from the full time series. Each sub-sequence is defined as

$$\tilde{\mathbf{x}}_{1:\tilde{T}}^{(p)} = \mathbf{x}_{t_p:t_p+\tilde{T}},$$

where $t_p \in \{1, \ldots, T-\tilde{T}\}$ is selected randomly. These sub-sequences are arranged into batches of size S, with each sub-sequence aligned to the corresponding trial-dependent weight matrices $\mathbf{W}_1^{(k)}$ and $\mathbf{W}_2^{(k)}$.

Adaptive Gaussian Noise. To enhance model robustness, adaptive Gaussian noise is added to each sub-sequence $\tilde{\mathbf{x}}_{1:\tilde{T}}^{(p)}$. For each sub-sequence, the standard deviation is computed along the specified dimension, and noise is drawn from a standard normal distribution. This noise is then scaled by both a noise level parameter η and the computed standard deviation.

Teacher Forcing Initialization. For each sub-sequence in the batch, an initial teacher signal is generated by mapping the observed data to the latent space via Equation 15. Specifically, the initial latent state is set as

$$\hat{\mathbf{z}}_t^{(p)} = \mathbf{I}\,\tilde{\mathbf{x}}_t^{(p)} \quad \text{for all } t,$$

which provides a data-driven initialization for the latent dynamics.

State Propagation. The initial latent state $\hat{\mathbf{z}}_1^{(p)}$ is used to initialize the model, and subsequent latent states are propagated according to the model dynamics:

$$\mathbf{z}_t^{(p)} = F_\theta\big(\tilde{\mathbf{z}}_{t-1}^{(p)}\big),$$

where F_θ denotes the nonlinear update defined by the model equations.

Reconstruction and Loss. The latent state predictions are mapped back to the observation space as

$$\hat{\mathbf{x}}_t^{(p)} = \mathbf{I}\,\mathbf{z}_t^{(p)}.$$

The reconstruction loss is quantified by the mean squared error (MSE) between the true sub-sequences $\{\tilde{\mathbf{x}}_t^{(p)}\}$ and the predicted sub-sequences $\{\hat{\mathbf{x}}_t^{(p)}\}$:

$$L_{\text{MSE}} = \frac{1}{S(\tilde{T}-1)} \sum_{p=1}^{S} \sum_{t=2}^{\tilde{T}} \left\| \tilde{\mathbf{x}}_t^{(p)} - \hat{\mathbf{x}}_t^{(p)} \right\|^2.$$

Optimization and Regularization. The loss function L_{MSE} is minimized via BPTT [253] using the RAdam [134] optimizer with an initial learning rate of 10^{-3}, decayed to 10^{-4} after 80% of the training epochs. Regularization terms are incorporated to enforce smoothness and continuity in the trial-dependent connectivity matrices. Specifically, the regularization loss for these matrices is defined as

$$\begin{aligned} L_W = &\frac{\lambda_1}{2K} \sum_{k=1}^{K} \left(\|\mathbf{W}_1^{(k)}\|_F^2 + \|\mathbf{W}_2^{(k)}\|_F^2 \right) \\ &+ \frac{\lambda_2}{2K} \sum_{k=2}^{K} \left(\|\mathbf{W}_1^{(k)} - \mathbf{W}_1^{(k-1)}\|_F^2 + \|\mathbf{W}_2^{(k)} - \mathbf{W}_2^{(k-1)}\|_F^2 \right) \\ &+ \frac{\lambda_3}{2K} \sum_{k=3}^{K} \Big(\|\mathbf{W}_1^{(k)} - 2\mathbf{W}_1^{(k-1)} + \mathbf{W}_1^{(k-2)}\|_F^2 \\ &\qquad + \|\mathbf{W}_2^{(k)} - 2\mathbf{W}_2^{(k-1)} + \mathbf{W}_2^{(k-2)}\|_F^2 \Big). \end{aligned}$$

The diagonal matrix $\mathbf{A}$ is regularized toward the identity matrix using an L_2 penalty:

$$L_A = \lambda_1 \|\mathbf{A} - \mathbf{I}\|_F^2 .$$

Standard L2 regularization is also applied to the bias terms $\mathbf{h}_1$, $\mathbf{h}_2$, and the input matrix $\mathbf{C}$:

$$L_{\text{weights}} = \lambda_1 \left(\|\mathbf{h}_1\|_2^2 + \|\mathbf{h}_2\|_2^2 + \|\mathbf{C}\|_F^2 \right).$$

The total regularization loss is given by

$$L_{\text{reg}} = L_W + L_A + L_{\text{weights}},$$

which is combined with the reconstruction loss to form the overall objective function minimized during training.

External Inputs

To provide the model with information about external influences that can result in perturbations of the autonomous dynamics reconstructed by the model, an external input matrix $S^(k)$ was specifically designed for each data set and trial segment. The specific design for each experiment can be found in Section 5.

Trajectory Generation from the Trained pePLRNN

To simulate latent trajectories from a trained pePLRNN, the model is initialized with a latent state $\mathbf{z}_0 \in \mathbb{R}^M$, an external input matrix $\mathbf{S} = [\mathbf{s}_0, \ldots, \mathbf{s}_T]^\top \in \mathbb{R}^{T \times U}$, and a time $T \in \mathbb{N}$. For each time step $t = 0, \ldots, T-1$, the latent state is recursively updated using Equation 15, with model parameters Θ fixed to the trial-specific set corresponding to the desired condition.

The resulting latent trajectory $\mathbf{Z} = [\mathbf{z}_0, \ldots, \mathbf{z}_T]^\top \in \mathbb{R}^{T \times M}$ evolves under the influence of the external inputs $\mathbf{S}$, starting from the specified initial condition $\mathbf{z}_0$. The observable trajectory $\mathbf{X} = [\mathbf{x}_0, \ldots, \mathbf{x}_T]^\top$ is then obtained via the observation equation 16.

Detailed conditions for each experiment on initial conditions, simulation length, and external input design are provided in Section 5.

2.3 Artificial Rule-Learning Task

I developed an artificial rule-learning task that mirrors the sequential structure of the animal's behavioral task as an artificial sequence-to-sequence task. I implemented two distinct versions of the original behavioral task: one that mimics the exact structure of the task (the cue remains present until the choice is made) and another that requires maintaining a memory of the cue by introducing a delay period. Each task variant distinguishes between the two rule conditions, denoted as the visual (VR) and the SR, and incorporates a parameter (fixing) to control whether the network is explicitly instructed to maintain a representation during the stimulus or delay phase.

Temporal Structure and Binning. Each trial is discretized into time bins, where one bin represents 50 ms in real time (to match the sampling of the neural data used for reconstruction). The task is segmented into several phases:

1. **Start Phase:** A baseline period lasting 60 bins (3 seconds) at beginning of each trial.
2. **Cue/Sample Phase:** In the version replicating the exact task structure, a cue is presented for 60 bins (3 seconds). In the memory variant, the cue is presented for a shorter period of 20 bins (1 second).
3. **Memory Phase (Memory Variant Only):** An additional phase of 100 bins (5 seconds) is introduced, during which the cue must be maintained in working memory.
4. **Choice Phase:** In the exact structure, the choice phase is randomly drawn from a specified range (10 to 110 bins). In the memory variant, the decision phase is shorter, varied randomly (e.g., 20 to 50 bins).
5. **Reward Phase:** A reward cue is presented over 10 bins (0.5 seconds).
6. **Reset Phase:** A final period of 60 bins (3 seconds).

In both task versions, the total trial duration is 300 bins (15 seconds).

Task Variants and Rule Conditions. Two primary task variants are used:

- **Exact Task Structure:** In this variant, the cue is continuously present from the cue onset the until the end of the choice phase.
- **Memory-Dependent Task:** This variant introduces a delay phase between the cue and choice phases, thereby requiring to maintain a memory of the cue information during the delay period.

Within each variant, there are three types of trials, two for the VR and one for the SR:

- **VR:** In VR trials, the cue information (e.g., left or right visual stimulus) is associated with a specific network output label during the choice phase (1 or 2). In all other trial phases, the output label is required to be 0.
- **SR:** In SR trials, there is only one output label during the choice phase disregarding the cue input (1). Again in all other trial phases the required network output label is 0.

An additional parameter, *fixing*, is used to explicitly require an output during the stimulus or delay phase of the task. In the case of the VR, the network is required to maintain a stimulus-specific output: trials with a right-cue are labeled with output 3, and trials with a left-cue are labeled with output 4 during the delay period. In contrast, under the SR, both cue conditions are labeled identically with output channel 3, regardless of cue identity.

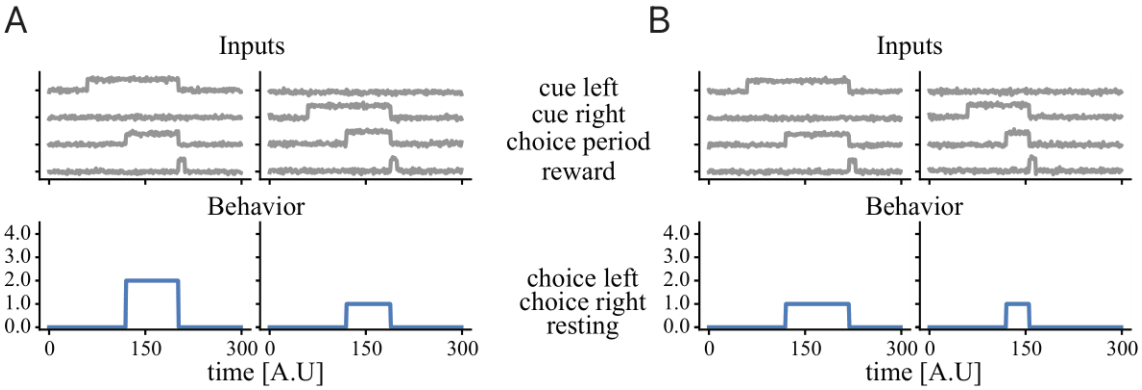

Figure 7: **Artificial task structure example A** Example of the temporal structure of two VR trials (left cue [left panel] and right cue [right panel]). External inputs (cue, choice reward signals) and required behavioral output (rest and choice) are aligned. During trials of the VR the required behavioral outputs during the choice phase are different (left cue requires output of 2 and right cue an output of 1). **B** Illustration of the temporal structure of two example trials during SR (left cue [left panel] and right cue [right panel]). Same as in **A** inputs and required behavioral output are aligned. During SR trials the required behavioral output is same, regardless of the input cue.

Randomness and Omission Trials I incorporate several sources of randomness:

- **Phase Duration Variability:** The durations of the decision phase in the exact task, and both the memory and decision phases in the memory variant are randomly sampled for each trial within a predefined ranges (see above).
- **Omission Trials:** With a probability of 33%, a trial is designated as an omission trial, in which no cue is presented and no decision is required. In these trials, the label for the entire trial is set to 0.

Input and Label Encoding Each trial is represented as a multidimensional array, where each time bin contains four input channels corresponding to specific task events:

- **Cue Channels:** Two channels encode the cue (left and right). Activation of channel 1 indicates a left cue, while channel 2 indicates a right cue.
- **Decision Cue Channel:** Channel 3 signals the decision period.
- **Reward Cue Channel:** Channel 4 indicates the reward period.

Additive Gaussian noise $\varepsilon_t \sim \mathcal{N}(0, 0.1\mathbf{I})$ was applied to the input.

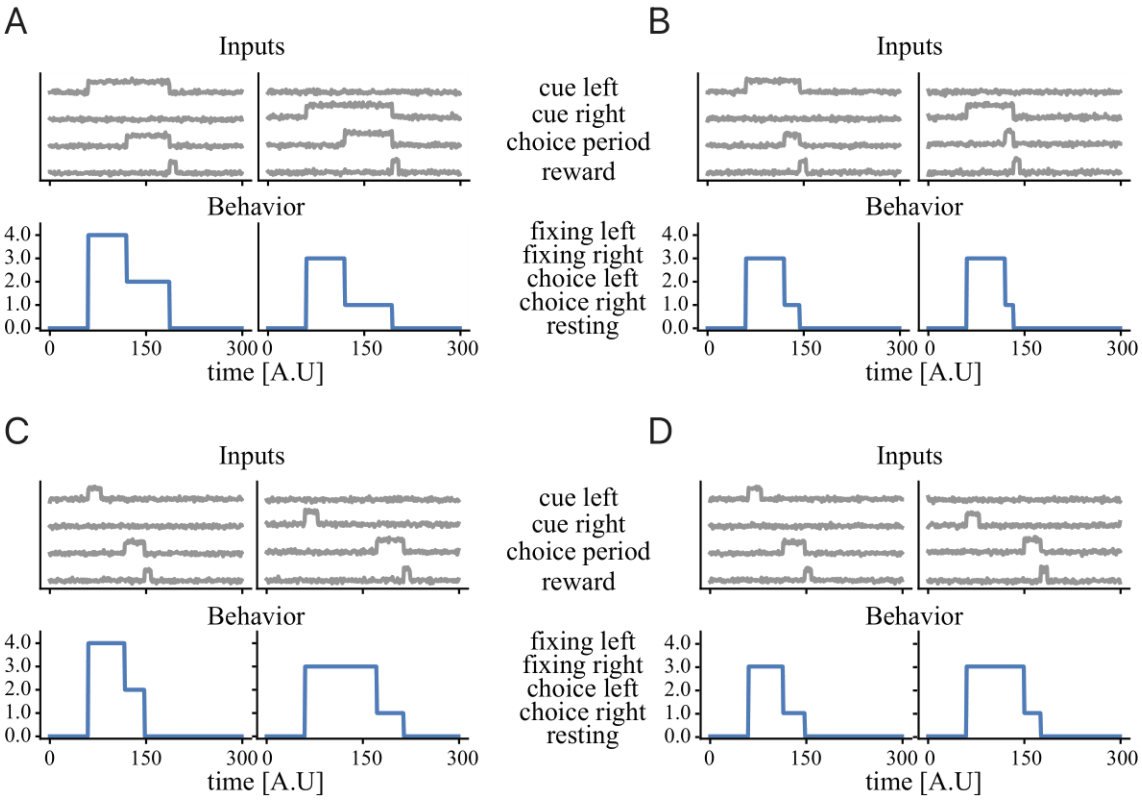

Figure 8: **Artificial task structure for fixing and memory (A, B)** Temporal structure example of two VR (**A**) and two SR (**B**) trials (for left cue and right cue). External inputs (cue, choice reward signals) and required behavioral output (rest and choice) are aligned. VR and SR trials require an additional fixing output after cue onset for the final choice. The required choice phase outputs are different depending on the cue (left cue requires output of 2 and right cue an output of 1). **(C, D)** Conceptually similar illustration as in **A** and **B**. But for trials of the task variant requiring memory. **C** depicts and example of VR trials (left and right) and **D** an example of SR trials. Here the cue input is only given for 20 time bins instead of the continuous input until the end of the choice phase.

2.4 The Task-Trained RNN

To model learning of the artificial rule-learning task, I implemented a simple RNN that maps the sequential inputs to the desired output labels as described above (see Sect. 2.3). The model is composed of two main modules: a one-layer RNN that processes the temporal sequence and a fully connected shallow decoder that produces the final output.

I used a standard one-layer vanilla RNN with tanh activation. At each time step t, the hidden state is updated as

$$\mathbf{h}_t = \tanh\left(\mathbf{W}_{\text{ih}}\,\mathbf{x}_t + \mathbf{b}_{\text{ih}} + \mathbf{W}_{\text{hh}}\,\mathbf{h}_{t-1} + \mathbf{b}_{\text{hh}}\right),$$

where $\mathbf{x}_t \in \mathbb{R}^{d_{\text{in}}}$ is the input vector, $\mathbf{h}_t \in \mathbb{R}^{d_{\text{hidden}}}$ is the hidden state, $\mathbf{W}_{\text{ih}} \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{in}}}$ is the input-to-hidden weight matrix, $\mathbf{W}_{\text{hh}} \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{hidden}}}$ is the hidden-to-hidden weight matrix, and $\mathbf{b}_{\text{ih}}, \mathbf{b}_{\text{hh}} \in \mathbb{R}^{d_{\text{hidden}}}$ are the respective bias terms.

The hidden state was initialized as

$$\mathbf{h}_0 \sim \mathcal{N}(0,\, 0.1^2\,\mathbf{I}).$$

The output of the RNN for each time step is then passed through a fully connected shallow decoder. The decoder computes

$$\hat{\mathbf{y}}_t = \mathbf{W}_2\,\text{ReLU}\left(\mathbf{W}_1\,\mathbf{h}_t + \mathbf{b}_1\right) + \mathbf{b}_2,$$

where $\hat{\mathbf{y}}_t$ is the predicted output at time t.

Sequential Training Procedure for the RNN The RNN model was trained sequentially on the two rule conditions, first the VR, then the SR, to simulate a learning process between the two rules.

Stage 1: VR. The network was first trained on VR trials by minimizing a total loss function composed of a cross-entropy loss over the output sequence and a regularization term applied to the hidden states. Let $\mathcal{L}_{\text{CE}}$ denote the cross-entropy loss and $\mathbf{h}_t \in \mathbb{R}^M$ the hidden state at time t. The regularization term was defined as

$$L_h = c \cdot \frac{1}{T} \sum_{t=1}^{T} \left[a \cdot \max\{-\mathbf{h}_t, 0\}^2 + b \cdot \max\{\mathbf{h}_t, 0\}^2\right],$$

with scalar coefficients $a = 10, b = 1, c = 10 \in \mathbb{R}_+$. The total loss was then

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + L_h.$$

All network parameters were optimized using the Adam optimizer with a learning rate of 10^{-4} and a weight decay of 10^{-6}, until the total loss $\mathcal{L} < 10^{-4}$ or a maximum of 5000 training epochs was reached.

Stage 2: SR. The second training phase began from the final parameter state of the VR training. To isolate the contribution of internal dynamics, only the RNN parameters associated with the hidden states ($\mathbf{W}_{\text{hh}}$ and $\mathbf{b}_{\text{hh}}$) were updated while keeping the decoder and input weights fixed. Training again minimized the same loss function $\mathcal{L}$, using the same optimizer and hyperparameters. Model training checkpoints were saved per training epoch to track the learning process of network adaptation from the VR to the SR.

2.5 Analysis Techniques

Behavioral Output Error for Task-Trained RNN and Reconstruction

I quantified the behavioral error of the task-trained RNN and its corresponding reconstruction model over all trials and time bins. For the reconstructed model, the original task-trained RNN's output decoder (see Sect. 2.4) was used to transform the reconstructed hidden state trajectories into the correct output space. For both models, the output vectors $\mathbf{o}_t \in \mathbb{R}^5$ were converted into class predictions by applying the softmax function followed by an argmax operation:

$$\hat{y}_t = \arg\max\left(\text{softmax}(\mathbf{o}_t)\right).$$

The behavioral error was then computed as the mean proportion of mismatches between predicted and true class labels across all trials and time points:

$$\text{Error} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{T_k}\sum_{t=1}^{T_k}\mathbb{I}[\hat{y}_t^{(k)} \neq y_t^{(k)}],$$

where K is the number of trials, T_k the number of time bins in trial k, and $\mathbb{I}$ the indicator function.

Linear Discriminant Analysis for Decoding

I used Linear Discriminant Analysis (LDA) for all decoding experiments as classification model. In all presented experiments LDA was always used in the binary case which justifies the following specifications. LDA estimates class-specific means μ_c and a common covariance matrix Σ, with set equal class priors $\pi_c = 0.5$ to address small sample sizes and potential class imbalance. The class-specific discriminant function is defined as:

$$\delta_c(x) = x^\top \Sigma^{-1}\mu_c - \frac{1}{2}\mu_c^\top \Sigma^{-1}\mu_c + \log \pi_c.$$

Each sample x is assigned to the class with the maximum discriminant score:

$$\hat{y} = \arg\max_c \delta_c(x).$$

Discriminant Vector The discriminant vector w determines the direction in the feature space that optimally separates the class means while minimizing within-class scatter. It is computed as:

$$w = \Sigma^{-1}(\mu_1 - \mu_2),$$

and defines the axis along which samples are projected to achieve maximal class separation.

Decision Boundary. In the binary case, the decision boundary is a hyperplane orthogonal to the discriminant vector w. This boundary is defined by the set of points x for which the two class scores are equal:

$$\delta_1(x) = \delta_2(x) \quad \Longrightarrow \quad w^\top x = \frac{1}{2} w^\top (\mu_1 + \mu_2).$$

The vector w determines the orientation of the boundary, and the intercept term depends on the average of the class means projected along w.

Discriminant Score For binary classification with classes $c = 1, 2$, the discriminant score represents the signed distance metric of a sample x from the decision boundary:

$$\delta(x) = w^\top x - \frac{1}{2} w^\top (\mu_1 + \mu_2).$$

A positive score assigns x to class 1, while a negative score assigns it to class 2. This scalar projection reduces high-dimensional state representations to a single discriminant axis, providing a principled metric for quantifying the degree and direction of class separation.

Cross-Validation for Classification Accuracy To assure statistical reliability of LDA-based classification results, I used a leave-one-out cross-validation procedure. Specifically, in each of $N = 100$ iterations, the dataset (X, y) was partitioned into a training and a test set leaving one sample out. Feature vectors in the training set were standardized, using the training mean μ_{train} and standard deviation σ_{train}:

$$X_{\text{scaled}} = \frac{X - \mu_{\text{train}}}{\sigma_{\text{train}}}.$$

LDA was then fitted using the scaled training data. The model's classification accuracy was computed on the test set and stored for that iteration.

Equal class priors $\pi_c = 0.5$ were used throughout. The final decoding accuracy was the mean accuracy over the N cross-validation iterations:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \text{score}_i.$$

Time-Resolved Decoding of Task Variables

To assess how neural representations of task variables evolved throughout the average trial in different rule contexts, I used a sliding window approach based on the cross-validated LDA described in Section 2.5. Sliding window decoding was restricted to trials from behaviorally stable periods of the visual and SR (see Methods **??** for exact trial definitions). Omission trials were excluded from the analysis.

Neural activity was averaged over a 5-bin sliding window, advancing in steps of one bin from time point $t = 20$ to $t = 200$. For each trial i, at time t, the neural state vector $\mathbf{x}_i(t) \in \mathbb{R}^N$ is:

$$\bar{\mathbf{x}}_{i,t} = \frac{1}{5} \sum_{k=0}^{4} \mathbf{x}_i(t+k),$$

At each window position t (for the group of considered behaviorally stable trials), I assessed the decoding accuracy of the following task variables:

- cue site (left vs. right),
- choice (left vs. right),
- reward outcome (delivered vs. omitted).

Cross-Validated Decoding of Task Variables in Stable Periods

To quantify the encoding of task-relevant variables in MSU activity during behaviorally stable trials, I used the cross-validated LDA framework described in Sections 2.5–2.5. Classification accuracy was evaluated for the three variables: cue site, choice, and reward outcome. All analyses were again restricted to stable performance periods of the visual and SR, excluding omission trials (see Methods **??**).

For each trial, neural activity was temporally averaged over the most informative time window as determined by the time-resolved decoding analysis in Section 2.5. Decoding accuracy was quantified for all included sessions.

Decoding Accuracy of Model-Generated Trajectories

To assess whether the pePLRNN-generated trajectories captured task-relevant neural firing rate patterns, I used the same cross-validated LDA framework described in Sections 2.5–2.5 to both recorded and model-generated neural trajectories. Classification accuracy was evaluated for cue site, choice, reward and rule type.

For this analysis, all trials were included. Neural state vectors were averaged over the same most informative time window determined by the time-resolved decoding analysis (Section 2.5). The time window used for the decoding analysis of rule type was the same as for the decoding analysis of choice. Decoding accuracy was computed for recorded and generated data across all sessions.

Robust Behavioral Decoding Framework

To enable a quantitative comparisons of neural states across rules in terms of their choice-related encoding, I developed a robust linear decoding framework based on LDA (see Sections 2.5) for behavioral classification.

The decoding task presented three principal challenges: (a) non-stationarity in the underlying neural activity across trials and rule periods, (b) a high-dimensional feature space due to the large number of recorded units relative to the limited number of behaviorally stable trials, and (c) a strong choice imbalance during the SR period, where only one choice remains reinforced.

To address these challenges, I trained an LDA on neural states of the choice phase during behaviorally stable trials of the VR, where both choices occur and are rewarded. This ensured that the decoder captured neural patterns associated with both left and right responses. The trained decoder was then tested on trials from the stable period of the SR, where only one response was rewarded.

To optimize the decoder's generalization across rules, I employed a stepwise unit elimination strategy. Let the full set of recorded units be denoted by

$$U = \{u_1, u_2, \ldots, u_d\}.$$

For each unit $u_j \in U$, a reduced subset was created by removing that unit:

$$U_{-j} = U \setminus \{u_j\}.$$

The decoding accuracy $A(U_{-j})$ was then computed for each subset. The unit u_{j^*} whose removal was associated with the greatest increase in decoding accuracy was identified as:

$$j^* = \arg\max_j A(U_{-j}).$$

This unit was removed from the set for the next iteration:

$$U \leftarrow U \setminus \{u_{j^*}\},$$

this procedure was repeated until only two units were left. At each step, decoding accuracy was tracked to identify the optimal subset post-hoc:

$$U_{\text{optimal}} \subseteq U,$$

which maximized decoding accuracy across both rule conditions. In cases of equal accuracy, the larger subset was preferred to retain maximal neural coverage.

State space analysis

Fixed point extraction of task-trained RNNs To extract fixed-points from trained task-trained RNNs, I used the fixed-point finder algorithm introduced by Golub et al. [93] (`https://github.com/mattgolub/fixed-point-finder`). This approach identifies points $\mathbf{x}^*$ in the network's state space that satisfy the condition

$$\mathbf{f}(\mathbf{x}^*) = \mathbf{x}^*,$$

where $\mathbf{f}(\cdot)$ denotes the recurrent update function of the RNN.

Candidate points for optimization were generated by sampling from the network's state space. Samples were drawn from actual hidden states encountered during task execution to ensure coverage of dynamically relevant regions. These candidates served as initialization points for the optimization procedure.

For each candidate $\mathbf{x}$, a fixed-point loss function was defined as

$$L(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{x}\|^2,$$

quantifying the deviation from the fixed-point condition. This loss was minimized using gradient-based optimizers (Adam or L-BFGS), and optimization proceeded until convergence or a predefined iteration limit was reached. A point $\mathbf{x}^*$ was accepted as a fixed point if $L(\mathbf{x}^*) < \epsilon$, with ϵ set to a small threshold.

To assess the local stability of identified fixed-points, the Jacobian matrix

$$\mathbf{J} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^*}$$

was computed at each converged location. The spectral radius of $\rho(\mathbf{J})$ was used to determine stability: fixed-points were classified as stable if $\rho(\mathbf{J}) < 1$, if $\rho(\mathbf{J}) > 1$ they were classified as unstable.

Fixed Point Extraction from the pePLRNN fixed-points $\mathbf{z}^* \in \mathbb{R}^M$ of the pePLRNN are defined by the condition $\mathbf{z}_{t+1} = \mathbf{z}_t = \mathbf{z}^*$, where the system dynamics follow:

$$\mathbf{z}_{t+1} = \mathbf{A}\mathbf{z}_t + \mathbf{W}_1^{(k)}\left[\varphi(\mathbf{W}_2^{(k)}\mathbf{z}_t + \mathbf{h}_2) - \varphi(\mathbf{W}_2^{(k)}\mathbf{z}_t)\right] + \mathbf{h}_1 + \mathbf{C}\mathbf{s}_t.$$

Due to the piecewise-linear nature of the ReLU nonlinearity $\varphi(x) = \max(0, x)$, all fixed-points were computed by exhaustively enumerating the 2^{2d} possible linear subregions of the latent space, characterized by binary diagonal matrices:

$$\mathbf{D}_1 = \operatorname{diag}\left[\varphi'(\mathbf{W}_1\mathbf{z}^* + \mathbf{h}_1)\right], \quad \mathbf{D}_2 = \operatorname{diag}\left[\varphi'(\mathbf{W}_1\mathbf{z}^*)\right].$$

Each pair $(\mathbf{D}_1, \mathbf{D}_2)$ defines a linear subregion in which the fixed point equation reduces to a solvable system:

$$\left[\mathbf{I} - (\mathbf{A} + \mathbf{W}_2(\mathbf{D}_1 - \mathbf{D}_2)\mathbf{W}_1)\right]\mathbf{z}^* = \mathbf{W}_2\mathbf{D}_1\mathbf{h}_1 + \mathbf{h}_2.$$

Solutions $\mathbf{z}^*$ were kept only if their subregion matched the original pair used to derive the solution.

The local stability of each fixed point was determined by evaluating the Jacobian matrix

$$\mathbf{J} = \mathbf{A} + \mathbf{W}_2(\mathbf{D}_1 - \mathbf{D}_2)\mathbf{W}_1$$

at $\mathbf{z}^*$. fixed-points were classified as stable if the spectral radius $\rho(\mathbf{J}) < 1$.

To examine input-driven attractor dynamics, fixed point extraction was repeated for a set of discrete external input vectors $\mathbf{s} \in \mathbb{R}^U$, each corresponding to a task-specific condition. For each input, the effective bias was modified as $\mathbf{h}_2' = \mathbf{h}_2 + \mathbf{C}\mathbf{s}$, and the fixed point computation was applied. The following external input configurations were used:

$$\begin{aligned}
\mathbf{s}_0 &= [0,0,0,0]^\top && \text{(baseline)} \\
\mathbf{s}_1 &= [1,0,0,0]^\top && \text{(cue right)} \\
\mathbf{s}_2 &= [0,1,0,0]^\top && \text{(cue left)} \\
\mathbf{s}_3 &= [1,0,1,0]^\top && \text{(response right)} \\
\mathbf{s}_4 &= [0,1,1,0]^\top && \text{(response left)} \\
\mathbf{s}_5 &= [0,0,0,1]^\top && \text{(reward)}
\end{aligned}$$

Vector Field Projection and Visualization To visualize the latent flow field of the pePLRNN model in a low-dimensional projection, I computed vector fields over a principal component plane fitted to a reference trajectory.

To define a projection plane, I applied principal component analysis (PCA) to the simulated trajectory $\{\mathbf{z}_t\}_{t=1}^{T}$, and extracted the first two principal components. A square meshgrid of size $n \times n$ was constructed in this 2D subspace, covering the region (x, y) with added margin scale, and then lifted back into the full latent space via inverse PCA transformation:

$$\mathbf{z}_{\text{grid}} = \text{PCA}^{-1}\left(\text{meshgrid}_{x,y}\right).$$

Each lifted grid point $\mathbf{z} \in \mathbb{R}^M$ was then evolved for one time-step using the pePLRNN dynamic equation 15.

The 2D first-order difference vectors between the projected original and advanced grid points was computed:

$$(\Delta x, \Delta y) = \text{PCA}(\mathbf{z}') - \text{PCA}(\mathbf{z}).$$

These differences define the direction and magnitude of the arrows of the flow field in the PC plane. For visualization, the vector lengths were normalized by

$$\eta(x, y) = \left(\Delta x^2 + \Delta y^2\right)^{-0.4},$$

and each component was rescaled as $\Delta x \leftarrow \eta \cdot \Delta x$, $\Delta y \leftarrow \eta \cdot \Delta y$, resulting in the normalized vector fields $U(x, y), V(x, y)$.

This procedure was used to generate flow field visualizations shown in Figure **??**.

Attractor State Space Visualization To visualize cue-specific attractor dynamics in section 3.2.3, latent state trajectories were projected onto a low-dimensional discriminant subspace derived from the robust choice decoder of the respective session (specified in 2.5). Let $\mathbf{W} \in \mathbb{R}^{N \times C}$ denote the matrix of LDA scalings, where N is the number of selected units and C the number of classes. Columns of $\mathbf{W}$ were orthonormalized using the Gram-Schmidt process to gain $\mathbf{U} \in \mathbb{R}^{N \times C}$, from which the first $d = 2$ columns defined the projection matrix $\mathbf{P} \in \mathbb{R}^{N \times d}$. For each trial type, trajectories $\mathbf{x}^{(i)}(t) \in \mathbb{R}^N$ were extracted over a fixed time interval $t \in [t_{\text{start}}, t_{\text{stop}}]$ and projected via $\mathbf{x}_{\text{proj}}^{(i)}(t) = \mathbf{P}^\top \mathbf{x}^{(i)}(t)$, yielding d-dimensional representations. A subset of trajectories was randomly selected and plotted together with mean trajectories $\bar{\mathbf{x}}_{\text{proj}}(t) = \frac{1}{R}\sum_{i=1}^{R} \mathbf{x}_{\text{proj}}^{(i)}(t)$.

Change Point Detection

PARCS PARCS model was used to detect CPs in time series via paired adaptive regression splines, following the framework of [240]. Data were first transformed using the cumulative sum (CUSUM) procedure, whereby the CUSUM-transformed time series $y = \{y_t\}_{t=1}^{T}$ was computed as

$$y_t = \sum_{\tau=1}^{t} \left(x_\tau - \langle x \rangle\right),$$

with $\langle x \rangle$ denoting the arithmetic mean of the original time series x. The PARCS algorithm then approximated y by a piecewise linear function whose bending points correspond to candidate CPs.

In a forward stage, spline pairs $h_+(c)$ and $h_-(c)$ centered at candidate CPs c were sequentially added to a linear regression model. The coefficients were estimated using least squares by minimizing the MSE:

$$\mathrm{MSE} = \frac{1}{T} \sum_{t=1}^{T} (y_t - \hat{y}_t)^2 .$$

A subsequent backward pruning procedure removed redundant spline pairs until a model with a predefined number M of CPs remained. These were then ranked according to their contribution to the explained variance.

To assess statistical significance, a block-permutation bootstrap procedure was applied. An H_0-conform time series x_0 was generated by regressing out the fitted PARCS model and inverting the CUSUM transformation. For each candidate CP c_m, a test statistic was defined as

$$S_m = \left| \hat{\beta}_{+m} + \hat{\beta}_{-m} \right| ,$$

quantifying the magnitude of bending at that point. B bootstrap samples were generated by permuting blocks of size k, preserving temporal dependencies. For each sample, the test statistic was recomputed, yielding an empirical distribution function (EDF). A CP was kept if its observed test statistic exceeded the $(1 - \alpha)$-quantile of the EDF; otherwise, it was rejected. The final model was refit using only significant CPs, and regression coefficients were re-estimated accordingly. PARCS was specifically used to detect one change point.

Sigmoidal Modeling of Behavioral Set Shifting I modeled binary choice behavior as a nonhomogeneous Bernoulli process with a time-varying success probability governed by a sigmoid. Each trial $i \in \{1, \ldots, N\}$ yielded a binary outcome $x_i \in \{0, 1\}$ drawn from Bernoulli$(s(t_i))$, where

$$s(t_i) = m + \frac{d}{1 + \exp\left(-\frac{t_i - c}{a}\right)}.$$

Here, m is the baseline rate, d the amplitude of the transition, a the inverse slope, and c the inflection point. This parametrization ensures $s(t_i) \in [0, 1]$ and captures abrupt shifts in behavior.

Parameters were estimated $\boldsymbol{p} = (m, d, a, c)$ by minimizing the negative log-likelihood:

$$\mathcal{L}(\boldsymbol{p}) = -\sum_{i=1}^{N} x_i \log s(t_i) + (1 - x_i) \log(1 - s(t_i)),$$

subject to the constraints:

$$0 \leq m \leq 1, \quad 0 \leq d \leq 1 - m, \quad a > 0, \quad 0 \leq c \leq 1.$$

MATLAB's `fmincon` with the SQP algorithm was used for constrained optimization. Initial values were set heuristically: c_0 was chosen as the point of maximal cumulative deviation from the mean response, m_0 and d_0 were derived from pre- and post-transition means, and $a_0 = 0.07$ (see Appendix for initial value evaluation, $a_0 = 0.07$ gave the highest average log-likelihood of $\mathcal{L}(\boldsymbol{p}) = 82.55$ for all datasets) was fixed.

Rule Bias Detection

I quantified the initial rule bias of animals at the beginning of experimental sessions through the analysis of choice probabilities during early trials. For each experimental dataset ($n = 18$), smoothed choice probability estimates were computed separately for left-cue and right-cue trial conditions. The estimation procedure involved:

$$P_{\text{smooth}}(a|s) = \mathcal{G} * \mathbf{1}_{a_t=a, s_t=s} \tag{17}$$

where $\mathcal{G}$ represents a Gaussian kernel with $\sigma = w/4$ (with window size $w = 6$), $\mathbf{1}$ is the indicator function identifying trials where action a was selected in stimulus condition s, and $*$ denotes the convolution operation.

Initial rule bias was classified by analyzing the first five trials in each cue condition. I computed the mean action probability matrix $\mathbf{P} \in \mathbb{R}^{2\times 2}$ where element $P_{s,a}$ represents the probability of selecting action $a \in \{\text{left}, \text{right}\}$ given stimulus $s \in \{\text{left-cue}, \text{right-cue}\}$:

$$\mathbf{P} = \begin{bmatrix} P(\text{left}|\text{left-cue}) & P(\text{right}|\text{left-cue}) \\ P(\text{left}|\text{right-cue}) & P(\text{right}|\text{right-cue}) \end{bmatrix} \tag{18}$$

Based on this matrix, initial biases were categorized into four distinct types:

- *Visual-rule biased*: $P(\text{left}|\text{left-cue}) > P(\text{right}|\text{left-cue})$ and $P(\text{left}|\text{right-cue}) < P(\text{right}|\text{right-cue})$
- *Right-rule biased*: $P(\text{left}|\text{left-cue}) < P(\text{right}|\text{left-cue})$ and $P(\text{left}|\text{right-cue}) < P(\text{right}|\text{right-cue})$
- *Left-rule biased* : $P(\text{left}|\text{left-cue}) > P(\text{right}|\text{left-cue})$ and $P(\text{left}|\text{right-cue}) > P(\text{right}|\text{right-cue})$
- *Confused/mixed strategy*: $P(\text{left}|\text{left-cue}) < P(\text{right}|\text{left-cue})$ and $P(\text{left}|\text{right-cue}) > P(\text{right}|\text{right-cue})$

These classifications were compared with the required initial rule to assess whether animals demonstrated rule-consistent biases.

Sliced Wasserstein Distance The sliced Wasserstein distance (SWD) was used to compare empirical distributions of latent state trajectories. Given two sets of samples $A, B \in \mathbb{R}^{n\times d}$, the SWD was computed by projecting both sets onto multiple random directions $\mathbf{v}_k \in \mathbb{S}^{d-1}$, and averaging the one-dimensional Wasserstein-1 distances across projections.

For each projection direction, the samples were sorted to compute the empirical quantiles, and the one-dimensional Wasserstein distance was computed as:

$$W_1(\hat{A}_k, \hat{B}_k) = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{A}_k^{(i)} - \hat{B}_k^{(i)} \right|,$$

where $\hat{A}_k = A\mathbf{v}_k$ and $\hat{B}_k = B\mathbf{v}_k$.

The final distance was obtained by averaging over all projection directions:

$$\text{SWD}(A, B) = \frac{1}{K} \sum_{k=1}^{K} W_1(\hat{A}_k, \hat{B}_k).$$

In the implementation, $K = 512$ projections were used, divided across four random samples of 128 directions each. Direction vectors were drawn from a standard normal distribution and normalized to unit norm. This method follows standard practices for estimating SWD as described in [21].

Effective Functional Connectivity Estimation To extract an estimates of effective functional connectivity from a trained model, latent trajectories are generated. For each time point along a trajectory, the local Jacobian $\frac{\partial z_{t+1}}{\partial z_t}$ is computed at time t taking the form:

$$J_t = A + W_1^{(k)} \operatorname{diag}\left[\varphi'(W_2^{(k)} z_t + h_2) - \varphi'(W_2^{(k)} z_t)\right] W_2^{(k)}. \tag{19}$$

Letting D_t denote the diagonal indicator matrix encoding the local subregion, the expression simplifies to:

$$J_t = A + W_1^{(k)} D_t W_2^{(k)}. \tag{20}$$

To capture the cumulative effect of local interactions across time, the time-ordered product of Jacobians is computed for each trajectory:

$$J_{\text{eff}} = \prod_{t=1}^{T} J_t. \tag{21}$$

This matrix summarizes the net influence of latent unit interactions across the trajectory. Averaging J_{eff} across multiple generated trajectories gives the estimate of effective functional connectivity for a specific condition and distribution of initial conditions:

$$\bar{J}_{\text{eff}} = \frac{1}{N} \sum_{i=1}^{N} J_{\text{eff}}^{(i)}. \tag{22}$$

To focus on the most dominant interaction patterns, I thresholded each $\bar{J}_{\text{eff}}$ by retaining the top 10% of absolute weight magnitudes (excluding diagonal elements). The resulting binary matrix was interpreted as a connectivity graph.

To compare effective functional connectivity structure across trials, I computed pairwise Jaccard similarity between thresholded adjacency graphs. For trials i and j, the similarity was defined as:

$$\text{Jaccard}(i, j) = \frac{|E_i \cap E_j|}{|E_i \cup E_j|}, \tag{23}$$

where E_i and E_j are the sets edges from trials i and j, respectively. This produced a symmetric similarity matrix for each session.

To test whether effective functional connectivity reorganized during learning, trial-wise similarity matrices were separated into clusters by the behavioral change point (CP) for each session. From each similarity matrix $S \in \mathbb{R}^{N \times N}$, I extracted two key cluster means based on the CP index c:

- The *pre–CP similarity* $\mu_{\text{pre-CP}}$, computed as the mean similarity between all trials before and after the CP (i.e., $i < c$, $j \geq c$),
- The *post–CP similarity* $\mu_{\text{post-CP}}$, calculated as the mean of the similarity for all trials occurring after the CP (i.e., $i, j \geq c$ and $j > i$).

2.6 Statistical Methods

Paired Statistical Tests

To evaluate whether real-values data differed significantly between two conditions, I applied a two-sample (independent) t-test across trials or sessions. The null hypothesis assumed equal population means. The test statistic and corresponding p-value were computed using `scipy.stats.ttest_ind`, assuming unequal variances by default (`equal_var=False`).

Statistical Testing for Bounded Measures

Unless stated otherwise, all statistical comparisons involving bounded variables (i.e. decoding accuracies, correlation coefficients and similarity) were performed using the Wilcoxon rank-sum test. This non-parametric test compares two independent samples drawn from distributions that may deviate from normality, particularly when the data are bounded within a fixed interval (e.g., [0, 1]).

The test was implemented using the `scipy.stats.ranksums` function in Python, which computes the rank-sum statistic and the associated p-value for the null hypothesis that both samples originate from the same distribution. A significance level of $\alpha = 0.05$ was used unless stated otheriwse.

All tests were two-sided unless otherwise noted. When reporting results, the test statistic and corresponding p-value are indicated in the text or figure captions where relevant.

Statistical Testing of Discriminability Values

To evaluate whether discriminant scores were significantly different from zero, I applied a one-sample paired t-test across trials or sessions. The null hypothesis assumed a population mean of zero. The test statistic and corresponding p-value were computed using `scipy.stats.ttest_1samp`.

Correlation Analysis of Reconstructed and Ground-Truth Trajectories

To assess the similarity between reconstructed and ground-truth trajectories, I computed trial-wise Pearson correlation coefficients across all latent or observed dimensions. This metric was used for neural data and task-trained RNNs.

Let $\mathbf{x}^{(k)} \in \mathbb{R}^{T\times N}$ and $\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^{T\times N}$ denote the true and reconstructed trajectories of trial k, where T is the number of time bins and N the number of units. For each unit j, the Pearson correlation coefficient between the true and reconstructed time series was computed:

$$r_j^{(k)} = \mathrm{corr}\left(\mathbf{x}_j^{(k)}, \hat{\mathbf{x}}_j^{(k)}\right).$$

The trial-wise mean correlation was then obtained by averaging over all units:

$$r^{(k)} = \frac{1}{N}\sum_{j=1}^{N} r_j^{(k)}.$$

To compare between distributions of datasets the mean across trials was calculated to obtain a mean correlation value per dataset.

Shuffle Distribution Generation

To generate the shuffled reference distribution in Section 3.2.4, CPs were randomly sampled across the trial sequence. For each shuffled configuration, block similarity was computed using the same procedure described in Section 2.5.

3 Results

The Results section is divided into two main parts. In the first part, I introduce the pePLRNN as framework for reconstructing non-stationary DS directly from data. I test this approach by reconstructing benchmark DS with parameter non-stationarity. This includes the logistic map and the bursting neuron model (s. Methods5, [55]), two examples of DS which undergo bifurcations as their parameters change. I then use the pePLRNN to reconstruct hidden state trajectories of task-trained RNNs trained on a sequence-to-sequence version of the rule-learning task performed by the animals (see Methods2.3), to further validate the modeling approach and show how a computational dynamic mechanism can be extracted from the trained model. I then show how specific experimental design choices can affect the computational dynamic mechanism used by task-trained RNNs to solve the rule-learning task. Therefor, two task variants are considered: one directly mirroring the conditions of the animal experiment, and another explicitly requiring the use of memory to solve the task. For each variant, I used the pePLRNN to reconstruct the hidden state dynamics of task-trained RNNs. I then use the trained pePLRNN as surrogate model to extract and characterize distinct components of the computational dynamic mechanism. I further show with the two task variants in a cross-condition experiment how reconstruction outcomes are influenced by system properties and experimental assumptions, specifically the external input design matrix provided to the model.

In the second part, I then use the pePLRNN to reconstruct MSU recordings from the prelimbic mPFC of rats performing the rule-learning task (see Methods2.1). The trained models successfully reconstruct neural firing rate profiles for both rules and capture key characteristics of the neural recordings and the animal's behavior. I first employed the trained models as surrogate systems for the underlying neural dynamics to investigate the computational mechanisms used during stable task performance periods (see Methods2.1). Analysis of the stable periods under each rule showed a monostable, input-driven mechanism underlying the decision-making process of the task. Each rule has a distinct pattern of stimulus-dependent attracting regions that guide the neural trajectories toward the correct behavioral response. With simulations I confirmed that, in the absence of external inputs, the autonomous system is monostable, exhibiting no evidence of multistability. This indicates that task-performance relies on stimulus-dependent attracting regions rather than intrinsic multistable attractor system. Building upon the analysis of stable performance periods, I then investigated how stimulus-dependent attractors, model parameters, and neural activity changed on a trial-by-trial basis during learning. All three quantities exhibited abrupt transitions, indicating shifts in the underlying neural dynamics. CPs extracted from the model dynamics consistently preceded behavioral CPs. To further characterize the evolving functional organization of units, I derived a method to estimate trial- and context-specific functional connectivity by simulating network activity and analyzing the resulting state-dependent weight matrices. This approach revealed that functional connectivity patterns formed distinct similarity clusters aligned with behavioral transitions, and that these functional connectivity patterns themselves underwent abrupt changes during learning.

3.1 Reconstruction of Benchmark Systems and Task-Trained RNNs

The Parameter-Evolving PLRNN

Modeling non-autonomous DSs from empirical non-stationary time series requires capturing time-dependent changes in the underlying vector field. Classical DSR approaches typically assume fixed parameters, resulting in a time-invariant vector field. To track non-stationarity, the current approach approximates non-autonomous system dynamics by allowing the model parameters to change across discrete temporal segments. Specifically, the observed time series is divided into segments with a distinct parameter set assigned to it, I call this parameter set *snapshot parameters*. This piecewise constant parameterization enables the model to locally approximate the DS with a *snapshot vector field*, which is a autonomous DS itself. The resulting framework is designed to captures temporal changes in the vector field by tracking time-dependent parameter reconfigurations. The model specified in 15 provides a tractable and interpretable method to approximate non-autonomous systems, specifically with parameter non-stationarity. These snapshot parameters are regularized with sparsity regularization $lambda_1$ and a continuity prior $lambda_2$. The continuity prior effectively regulate how much change in parameters is allowed for consecutive temporal segments.

Reconstruction of Benchmark Systems

To validate the ability of pePLRNN to reconstruct DS with parameter non-stationarity, I first evaluated the model's reconstruction performance on two benchmark systems: the logistic map and a bursting neuron model ([55]). The logistic map is a one-dimensional discrete system exhibiting a sequence of bifurcations as its control parameter varies, going from a stable fixed-points to periodic oscillations and finally to chaotic behavior. In contrast, the bursting neuron model represents a continuous-time system with multiple intrinsic time scales, producing fast spiking activity but also slower oscillations. I chose these systems because of their simplicity and the fact that both exhibit well-known bifurcations, providing a ground-truth test cases for the model's ability to capture multiple abrupt qualitative shifts in system behavior as well as multiple time scales.

The model reconstructs both systems with their the main dynamic features. For the bursting neuron model and the logistic map, pePLRNN accurately captures the distinct dynamic regimes and time scales (Figure 9A-B).

The bifurcation diagram reconstructed from the pePLRNN matches the ground-truth bifurcation diagram of the logistic map, accurately capturing the transition points of the different dynamic regimes (Figure 9C)

To reflect parametric non-stationarity, the pePLRNN was trained (see Sect. 2.2) simultaneously across multiple dynamical regimes, with each regime assigned a distinct set of connectivity matrices $W_1^{(k)}$ and $W_2^{(k)}$ linked to specific segments of the time series.

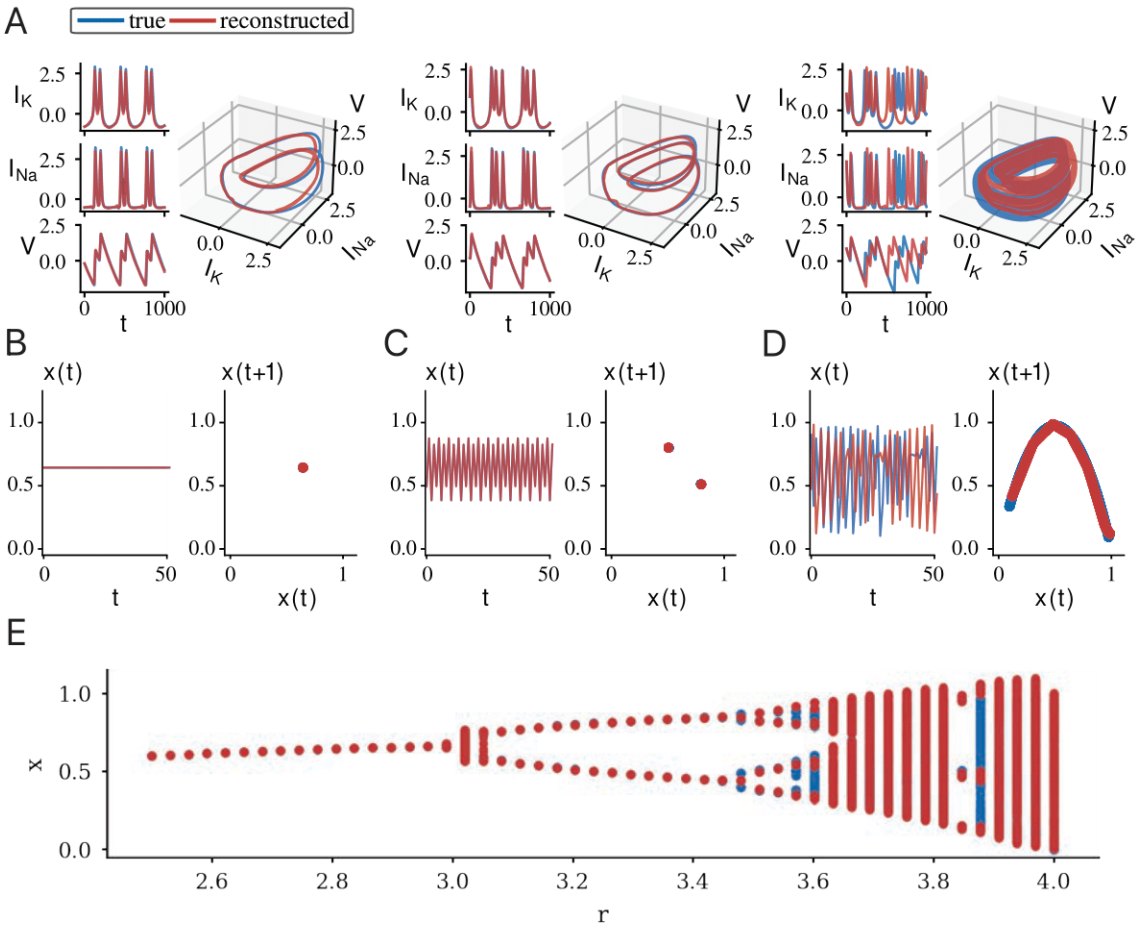

Figure 9: **Benchmark system reconstructions demonstrate the capacity of the pePLRNN to recover dynamic mechanisms with parameter non-stationarity. A** Reconstruction of the bursting neuron model under three distinct dynamical regimes: quiescent, bursting, and tonic firing. Blue traces represent the ground-truth simulated system, while red traces show the pePLRNN reconstruction. Each regime is characterized by distinct current dynamics (I_K, I_{Na}) and membrane potentials (V), which are faithfully captured by the model. **B** Reconstruction of the logistic map under three parameter settings corresponding to fixed-point, periodic, and chaotic behavior. Ground-truth trajectories (blue) and reconstructed trajectories (red) closely align across different regimes. **C** Bifurcation diagram of the logistic map, illustrating the correspondence between true and reconstructed bifurcation structures across a range of control parameter values. The reconstructed bifurcation pattern (red) accurately tracks the emergence of periodic windows and the onset of chaos observed in the ground-truth system (blue).

3.1.1 Reconstruction of Rule-Learning Task-Trained RNNs

To further validate the models ability to reconstruct the non-stationary neural dynamics underlying rule learning, I used the pePLRNN to reconstruct the hidden state trajectories from simple RNNs trained on an artificial version of the animal's rule-learning task (Figure 10). In the artificial task, the simple RNN received structured input sequences representing cue presentation, decision period, and reward feedback, and was trained to generate the corresponding output sequences encoding behavioral responses. The two rules were implemented by requiring different output sequences from the RNN for each rule. To simulate rule learning, the simple RNN was first trained on the VR paradigm with all parameters free to train. After learning the VR, the trained RNN was retrained on the SR, this time allowing only parameters associated with hidden states to be updated, while all other parameters were fixed (see Methods 2.4 for details). This ensured that learning the SR was implemented completely via changes in hidden state dynamics.

After training, hidden state trajectories of the task-trained RNNs were sampled during the VR, learning period between VR and SR, and the fully trained SR phases (see Methods for exact details). Hidden state trajectories together with the associated inputs were used to train pePLRNNs to reconstruct the hidden state dynamics of the task-trained RNNs (see Figure 10A).

The pePLRNN accurately reconstructed the hidden dynamics underlying the rule-learning task in the simple RNN, for both stable rule phases as well as during the learning period (see Figure 10B for two example trials from the test set). Validation on newly generated artificial trials showed high correlations between true and reconstructed trajectories (Figure 10F), demonstrating robust generalization on unseen data. In addition, fixed-points extracted from the reconstructed system closely matched those of the original task-trianed RNN across all stable and learning conditions (see Figure 10C). The pePLRNN reconstructed dynamic objects of the original system across all conditions without being explicitly trained on them. The model also preserved the original system's geometry, as illustrated by the example in full state space, generated from unseen test data (Figure 10D). Behavioral outputs decoded from the reconstructed dynamics matched those of the original task-trained RNN without any direct training on task labels (Figure 10E). Together, this validated the model reconstruction as a functional surrogate and shows that the approach works.

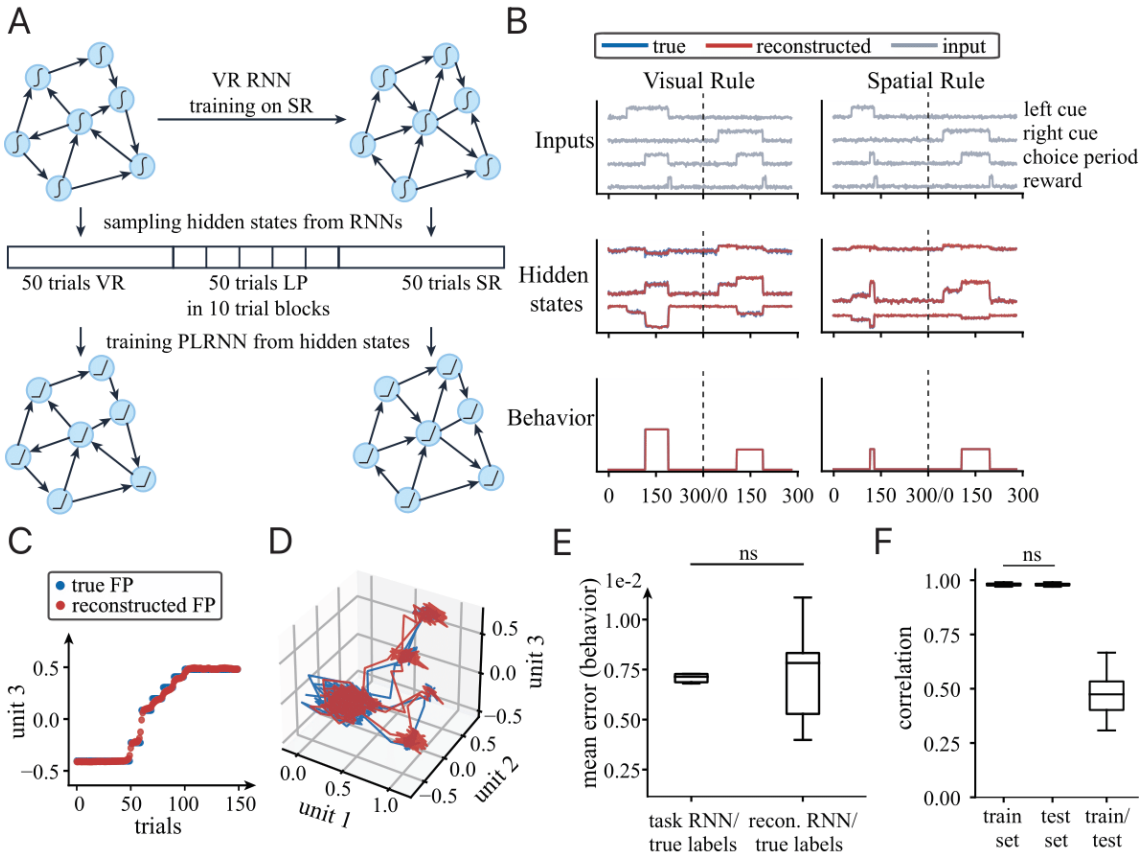

Figure 10: **Reconstruction of hidden state dynamics from an artificial rule-learning task.** **A** Illustration of the training protocol: the vanilla RNN was first trained on the VR, then retrained over a learning period on the SR. Hidden states were sampled from task-trained RNN trial simulations of VR, learning period, and SR phases. These hidden state data sampel served as dataset to train the pePLRNN. **B** Reconstruction performance on the test set: comparison of true (blue) and reconstructed (red) hidden state trajectories. Top row shows task inputs including cue presentation, choice period and reward. Middle row displays hidden state trajectories, demonstrating close agreement between true and reconstructed dynamics. Bottom row depicts behavioral outputs. Vertical dashed lines separated two distinct trials. **C** Comparison of fixed-points extracted from true and reconstructed systems, showing close overlay over all phases. **D** Three-dimensional (full) state space representation of true and reconstructed trajectories during left- and right-cued trials. **E** Mean behavioral prediction error shows comparable decoding performance between the original RNN and the reconstructed system. Note that the scale is at $10^{-}2$ meaning both achieve over 99% behavioral accuracy during stable performance periods. **F** Correlation analysis for training, and test sets.

Extracting the Computational Dynamic Mechanism from the Reconstruction Model

After validating the model's ability to reconstruct the underlying system, I used the trained pePLRNN as a surrogate system to investigate the dynamic computational mechanisms generating the RNN's behavioral output. I introduced a case distinction based on the external inputs to analyze the dynamics in different trial phases.

In the first case, the RNN receives no external inputs and thus evolves autonomously. In the second and third cases, the cue input is activated, corresponding to the presence of left or right cues. In the fourth and fifth cases, both cue and choice period inputs are active, while in the sixth case, the reward input is active. These conditions are formally represented by six external input vectors S_i:

$$S_0 = [0,0,0,0]^\top, \quad S_1 = [1,0,0,0]^\top, \quad S_2 = [0,1,0,0]^\top,$$
$$S_3 = [1,0,1,0]^\top, \quad S_4 = [0,1,1,0]^\top, \quad S_5 = [0,0,0,1]^\top.$$

Each S_i can be interpreted as a new effective bias on the latent dynamics. With this, I interpret the input-driven cases as six distinct autonomous systems and analyze how the additional bias term influences the system attractor geometry and stability:

$$z_{t+1} = Az_t + W_1^{(k)}\left(\varphi\left(W_2^{(k)} z_t + h_2\right) - \varphi\left(W_2^{(k)} z_t\right)\right) + \begin{cases} h_1, & \text{if } s_t \approx S_0 \\ h_1 + CS_1, & \text{if } s_t \approx S_1 \\ h_1 + CS_2, & \text{if } s_t \approx S_2 \\ h_1 + CS_3, & \text{if } s_t \approx S_3 \\ h_1 + CS_4, & \text{if } s_t \approx S_4 \\ h_1 + CS_5, & \text{if } s_t \approx S_5 \end{cases} \tag{24}$$

For each case S_i, I extract the FPs exhaustively to obtain the full view on fixed point attractors of the system. For every case, the system exhibited exactly one stable fixed point acting as a global attractor. Depending on the external input configuration, the location of the stable fixed point changes within the latent state space, thereby guiding the trajectory toward task-relevant regions.

The behavioral output linked to each fixed point showed a clear functional distinction between the trial phases. In the autonomous phase (S_0), the system stays in a resting state and produces a neutral output (decoded as 0; see Figure **??**B). During the cue phases (S_1, S_2), the fixed point shifts to different regions in state space, but the behavioral output stays unchanged. The system produces only in the decision phases (S_3, S_4) distinct outputs, corresponding to left (2) and right (1) choice. Finally, in the reward phase (S_5), the trajectory is guided back toward the resting state, and the behavioral output returns to 0.

During the transition from visual to SR, the locations of the fixed-points changed across all input conditions (see Figure **??**C). Only in the choice-right case, a change in behavioral output decoding occurred - from a left choice to a right choice - correctly implementing the SR. Hence implementing the SR is realized in the RNN's dynamics by shifting the input-driven attractors in state space.

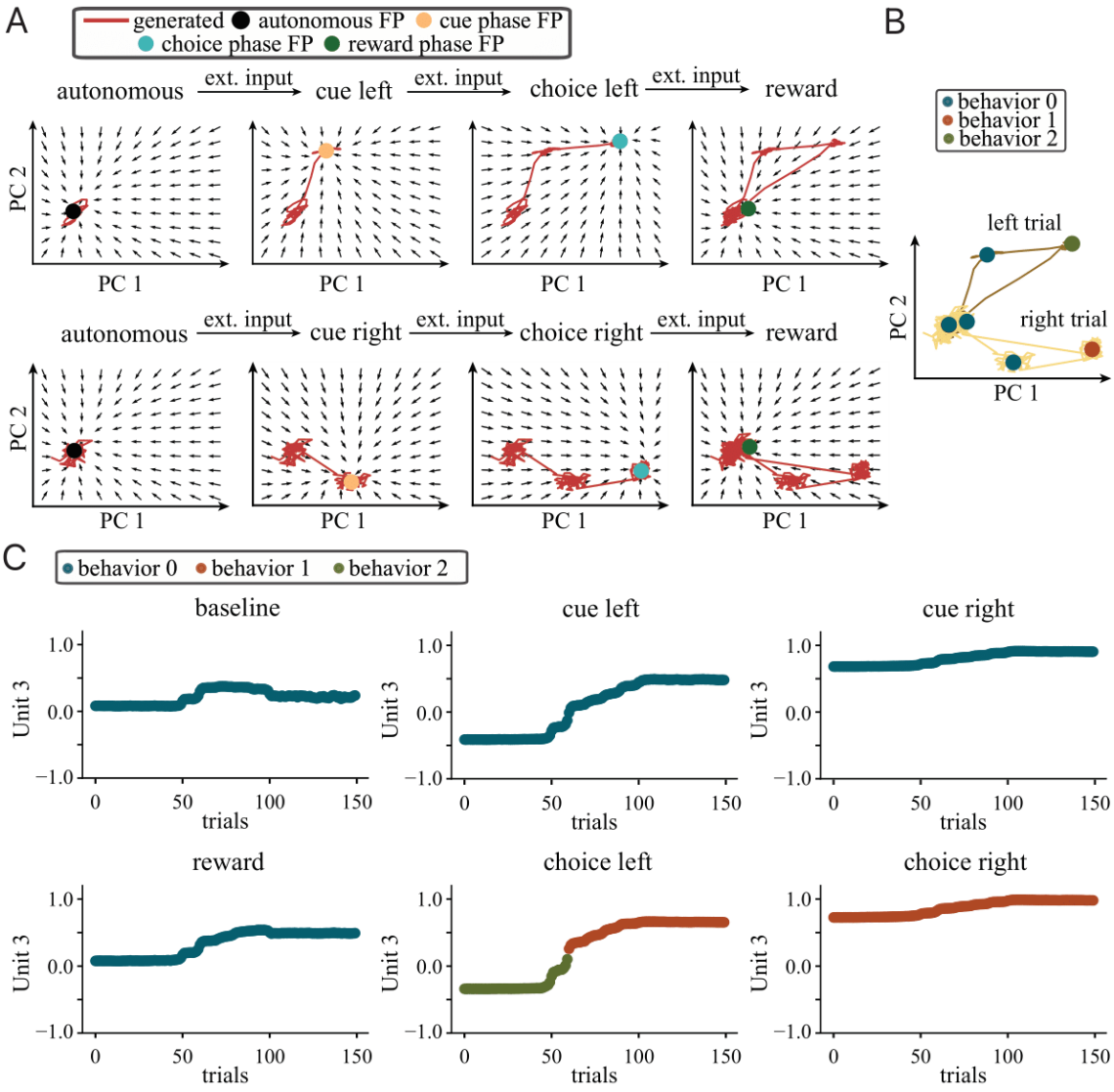

Figure 11: **Reconstructed computational dynamics of task-trained RNNs as input-driven fixed point transitions in state space. A** Flow fields of reconstructed dynamics projected onto first two principal components across different task stages. Top row depicts a left trial sequence, while bottom row shows a right trial sequence with corresponding FPs, and red lines show generated trajectories from the reconstructed model. The autonomous system (leftmost panels) has one FP marking the resting state (black). With external inputs corresponding to either left or right cue (second column), the system's dynamics change, creating a new fixed-points (orange) that attract trajectories in a cue-specific manner. During the choice phase (third column), the FP location changes again, creating choice-specific states (cyan). Finally, during the reward phase (rightmost panels), the FP location changes again guiding the trajectory back in the vicinity of the resting state FP. **B** Behavioral decoding (resting behavior: 0, right choice: 1 and left choice 2) of all FPs shows that only the choice-phase FP discriminate behavioral output.

Figure 11: **C** Trial-wise locations of fixed-points in latent space (depicted only for one unit) for each external input condition. Across all conditions, fixed-points shift during learning the SR. For all input conditions, except the choice-left condition, behavioral output decoding stays the same for all trials. Only for the choice-left condition the behavioral output decoding changes from left choice to right choice.

3.1.2 The Influence of Memory on the Computational Mechanism behind Rule Learning

To assess how task-specific features influence the underlying computational dynamics, I compared two variants of the original rule-learning task. In both cases, the network was required to produce a distinct output during the cue and choice phases, thereby requiring the network to keep internal representation of the cue. In the first task variant, a continuous cue input was presented until the end of the choice period, similar to the structure of the original animal task. The second variant introduced a memory component: the cue was presented briefly, followed by a delay period without input, after which the correct choice output had to be produced based on the stored cue.

Both variants were reliably learned by the task-trained RNNs (error rate $<1\%$), and the pePLRNN accurately reconstructed the hidden state dynamics for both variants. Reconstructed trajectories, behavioral outputs, and stable fixed point locations in the autonomous case showed high agreement with those of the original task-trained RNNs (Figure 12A–F), validating the reconstructed models as surrogate models.

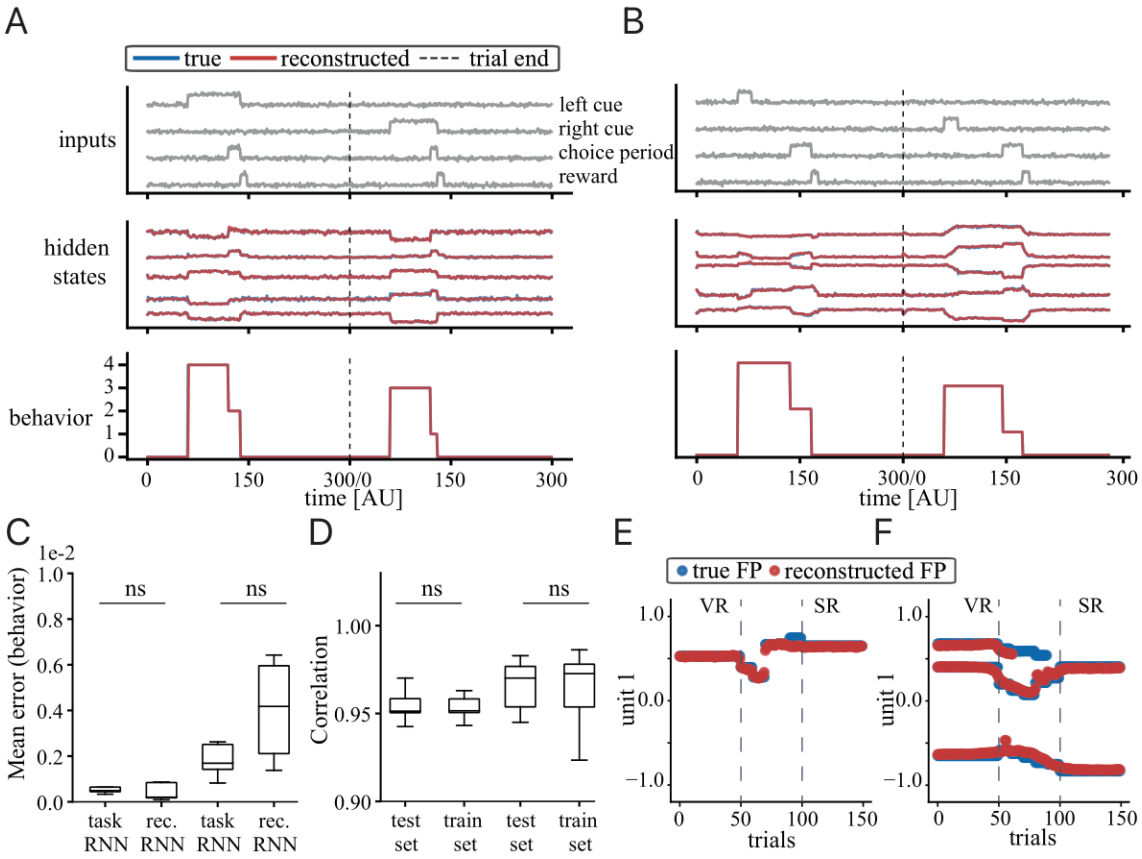

Figure 12: **Reconstruction of hidden dynamics and attractor structure across task variants with and without memory requirement. A** Example test trial from the task variant without memory. Top: input; middle: hidden state trajectories from the original task-trained RNN (blue) and pePLRNN reconstruction (red); bottom: behavioral outputs (neutral output: 0, cue right: 3, cue left: 4, right decision: 1, left decision: 2). Continuous cue input is present until the end of the decision phase. **B** Test trial example from the memory-dependent task variant. Cue input is presented only for 20 time bins, followed by a delay period with no inputs. The network has to maintain internal cue representation until the decision is required. **C** Behavioral error of the two task variants. Both task variants are learned with less than 1% error rate. Reconstructed models achieve comparable results as the original task-trained RNN. **D** Correlation comparison between the reconstructed hidden state trajectories and the reconstructed trajectories for both task variants for test and train set. There is no significant difference between the train set and the test set for both task variants. **E** Comparison of stable fixed-point locations between the task-trained RNN and pePLRNN reconstructions for the no-memory task. fixed-points are extracted from autonomous period. **F** Same as in E, but for the memory task. In both task conditions fixed-points from the reconstructed system align with the fixed-points extracted from the original task-trained RNN.

In the autonomous case, the attractor structure differs between the two task variants. During the VR, the continuous-input condition exhibits a single stable fixed point corresponding to a neutral behavioral output (0). In contrast, the memory-

dependent variant produces three stable fixed-points: one associated with neutral output and two representing internal cue representation, decoded as outputs 3 and 4 (Figure 13A–B). This difference is consistently observed across all trained RNNs and their reconstructions.

When the SR is learned, the attractor structure in the continuous-input variant stays qualitatively the same. The fixed point changes location over the course of learning, but there is no change in the number of fixed-points or the behavioral output decoding. In the memory variant the number of stable fixed-points decreases. The attractor encoding the cue that is no longer needed for solving the SR disappears during learning (Figure 13C–D). This change in number of attractors reflects a bifurcation during learning the SR.

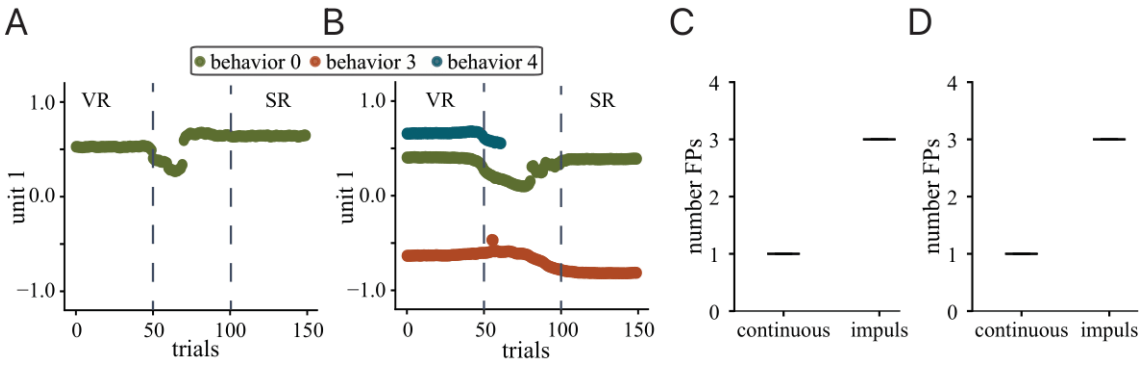

Figure 13: **Attractor structure behavioral decoding across trials for both task variants in the autonomous regime.** **A** Fixed point locations for the continuous-input task variant, projected onto the latent space and colored by behavioral output (resting: 0, internal cue 3 or 4). Only a single attractor associated with neutral output is present across trials. **B** Same as A, but for the memory-dependent task variant. Three distinct fixed-points are reconstructed under autonomous conditions: one reflecting the resting state and two corresponding to internal cue representations. After learning the SR the number of stable fixed-points decreased by one, marking a bifurcation event during learning. **C** Number of stable fixed-points extracted from the autonomous system across trials during the VR for the original RNN. **D** Same as C, but for the pePLRNN reconstruction. The reconstructed model reproduces the number of fixed-points in both task conditions correctly.

3.1.3 Input Design Matrix Influences Reconstruction Outcome

To examine how assumptions about the structure of external inputs influence the result of DS reconstruction, I performed a cross-condition analysis using the two variants of the rule-learning task introduced earlier. Here, I reconstructed the hidden state trajectories of the RNN trained on the continuous-input task providing impulse-based cue inputs (similar to the inputs from the memory task). In contrast, the hidden states of the RNN trained with impulse-cue inputs were reconstructed using continuous-cue inputs.

The cross-input reconstruction showed a clear qualitative differences in attractor structure depending on the input configuration used during training. When reconstructing the continuous-input RNN with impulse-based cue inputs, the model exhibited three stable fixed-points in the autonomous conditions, compared to only one in the original system (Figure 14A). The reconstructed attractor geometry matches more closely with the one of the memory task than the true continuous-input model. Conversely, when the impulse-trained RNN was reconstructed using continuous inputs, the resulting model recovered only a single fixed point during the VR, failing to reconstruct the three original attractors that encoded internal cue representations (Figure 14B). In both cases, the mismatch in attractor count was accompanied by a shift in behavioral decoding matching again the task variant with other input structure.

These structural changes were also reflected in other reconstruction metrics. For the continuous-input RNN, impulse-based reconstruction led to a significant increase in behavioral prediction error and a decrease in correlation between reconstructed and original hidden state trajectories (Figure 14C). In contrast, reconstruction of the impulse-trained RNN with continuous inputs showed no significant differences in these measures (Figure 14D).

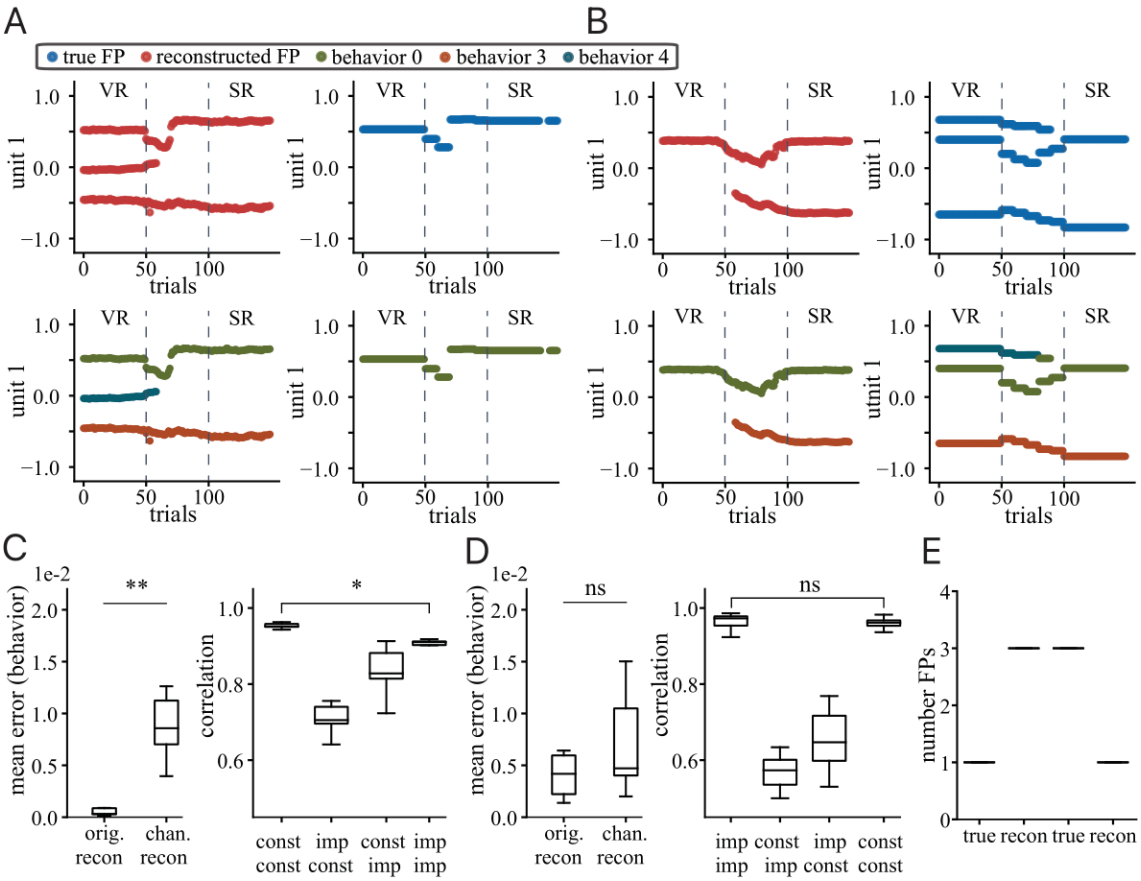

Figure 14: **Swapping input assumptions alters attractor structure and reconstruction quality. A** Reconstruction of hidden states from a continuous-input-trained RNN using impulse-like inputs. Left: reconstructed fixed-points (top) and corresponding behavioral decoding (bottom) over trials. Right: fixed-points (top) and behavioral decoding (bottom) from the original continuous-input task-trained RNN. The reconstruction shows three distinct attractors instead of one, and behavioral decoding matches the memory task profile. **B** Reconstruction of hidden states from an impulse-trained RNN using continuous inputs. Left: reconstructed fixed-points (top) and behavioral decoding (bottom). Right: corresponding ground-truth dynamics from the original impulse-trained RNN. The reconstruction shows a single fixed point during the VR instead of three. **C** Reconstruction error analysis for the continuous-input task. Left: mean behavioral prediction error increases significantly when reconstructed with mismatched impulse inputs ($p < 0.01$). Right: correlation between reconstructed and true hidden states for all input-model combinations (always on test data). The imp–imp (impuls inputs with reconstruction model trained with impulse inputs) condition shows a significant drop in correlation compared to the matched const–const condition ($p < 0.05$). **D** Same analysis as in C but for the memory task. Despite a structural mismatch in attractors, behavioral error and correlation remain stable across the two aligned input combinations. **E** Comparison of the number of stable fixed-points in the autonomous regime between the original and reconstructed models for each input condition. Differences in attractor count illustrate the strong influence of input assumptions on the reconstructed system's geometry.

3.2 Reconstructing the Neural Dynamics of Rule-Learning Rodents from Neural Measurements

3.2.1 Animal behavior

Six rats were trained in a probabilistic rule-shifting paradigm, a modified version of the task described by [74] (training and experiment conducted by Dr. Florian Bähner). Prior to the experimental recordings, animals were exclusively trained on the VR to establish baseline task performance. The paradigm required animals to flexibly switch between two rules: the previously learned VR and a novel SR. During VR, animals learned to associate visual cues (left or right cue light) with the correct lever press to obtain probabilistic rewards (80% reward probability for rule-aligned responses, 20% for inconsistent responses). After reaching a performance criterion of 80% correct responses over the last 20 trials, the rule was switched to the SR without explicitly providing a cue to the animals. In the SR, reinforcement depended only on the lever side, independent of the visual cue (Figure 17). In subsequent recording sessions, animals needed to perform alternating rule-switches to further test their cognitive flexibility. Sessions alternated between rule-switches from VR to SR and from SR back to VR. Animal learning behavior showed individual variability in switching between rules, learning speed, and persistence of learned rules (see Figure 16). Behavioral performance across 22 analyzed sessions showed that in 18 sessions, animals successfully transitioned between the VR and the SR. In four sessions the animal failed to switch from the SR to the VR performing only the SR, and one session in which the animal failed to switch from the VR to the SR (Figure 16A).

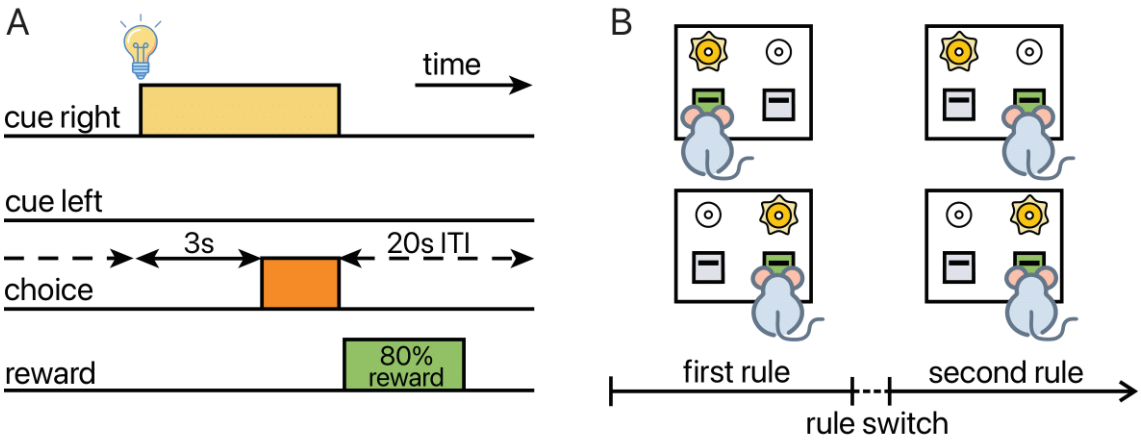

Figure 15: **Probabilistic rule-learning paradigm.** Each trial began with the presentation of a left or right cue light for 3 s, followed by a 10 s choice phase. If the animal did not make a choice within the 10 s time window, the trial was counted as an *omission*. Reward delivery followed the choice phase. Initially, the *first rule* (VR) determined reward outcomes; after reaching criterion, the task shifted to the *second rule* (SR) without explicit cueing.

Single trial behavioral performance for individual animals shows the abrupt transitions during rule learning. After the unannounced rule change, individual trial

outcomes (gray dots) fluctuated between correct and incorrect responses. After this learning period, behavioral performance showed an abrupt transition toward correct responses rather than a gradual improvement. Smoothed convolution of outcomes (gray trace) and the fitted sigmoidal behavioral model (black line) illustrate this behavior in Figure 16B for two examples. I used the sigmoidal behavioral model to extract these points of abrupt performance increase as behavioral CP. Comparing behavioral performances of different experimental phases confirmed these abrupt changes in behavior across sessions. Performance dropped significantly after the rule change ($p < 0.001$), then remained low during the learning period (before the behavioral CP), and increased significantly only after the behavioral CP ($p < 0.01$; Figure 16E).

Animals showed different individual performance trajectories based on their initial rule bias. To investigate this, I categorized sessions according to the initial response pattern of the animals (Figure 16D). Of the 18 valid sessions, nine started with a correct initial bias and nine with an incorrect bias (Figure 16F). Animals 3 and 5, started with the wrong initial bias in all their sessions, while others varied (Figure 16H). Animals with correct initial rule bias reached the rule-change performance criterion significantly faster than animals with incorrect initial bias (Figure 16C). After the rule change, however, the number of trials required to reach the behavioral CP was not significantly affected by the initial bias (Figure 16D). For most animals the behavioral CP occurs within 50 trials after the rule change (Figure 16G).

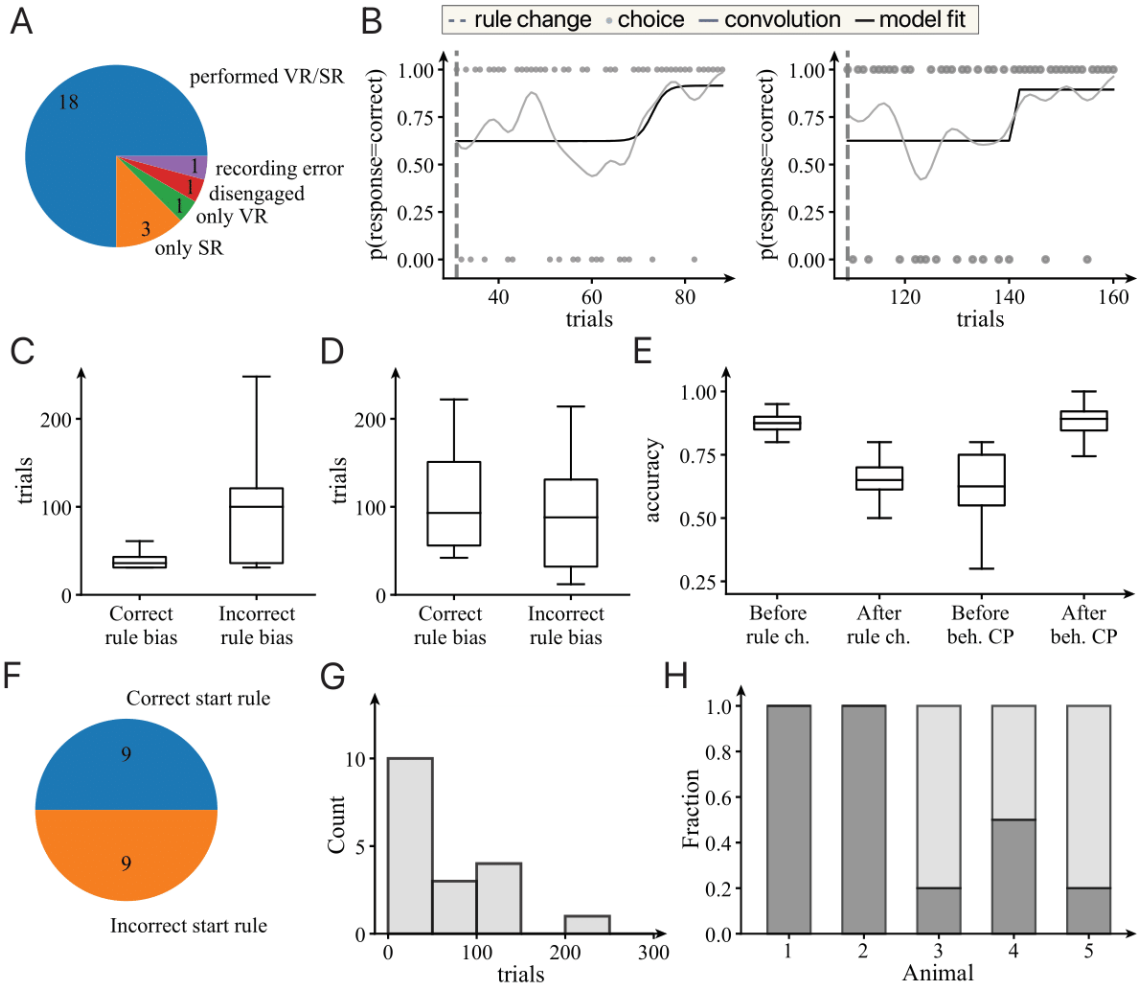

Figure 16: **Behavioral performance and learning dynamics during rule switching.** **(A)** Session outcomes categorized by rule performance. **(B)** Representative examples of trial-by-trial behavior during rule switching. Gray dots: individual trial outcomes; gray line: smoothed convolution; black line: fitted sigmoidal behavioral change point model. **(C)** Number of trials until rule-change performance criterion was reached, split by initial rule bias. **(D)** Number of trials to the behavioral change point after rule change, split by initial rule bias. **(E)** Accuracy comparisons across different experimental stages: before rule change, after rule change, before behavioral CP, and after behavioral CP. **(F)** Fraction of sessions with correct versus incorrect initial rule bias. **(G)** Distribution of learning periods until the behavioral change point was reached. **(H)** Animal-specific initial bias across sessions. Gery indicates correct rule bias and light grey incorrect rule bias.

Direct Neural Decoding and Robust Choice Decoding Framework

Before starting with model reconstructions, I analyzed the neural recordings directly to investigate differences in the encoding of task-relevant information between the visual and SR.

Decoding task-related labels from whole-trial average firing rates showed that most trial-specific variables, such as choice and reward, could be decoded equally

well during both the VR and the SR. In contrast, stimulus (cue site) decoding exhibited a strong rule-dependent difference: during the VR, stimulus identity could be robustly decoded from MSU activity, whereas during the SR, stimulus decoding dropped to chance level (Figure 17C).

To resolve the temporal structure of encoding during a trial, I computed decoding accuracies of stimulus and choice across all time bins (within a trial) during the stable performance periods. During the VR, decoding accuracy for both choice and stimulus rose after cue onset, peaked during the choice period (approximately 3–4 seconds after cue onset), and then declined again (Figure 17A). In contrast, during the SR, stimulus decoding remained at chance level throughout the trial, while choice information could be reliably decoded at all times, including before cue onset and after the choice phase (Figure 17B).

These results show how stimulus and choice encoding changes across rules: while both are encoded in the neural states during the VR, the stimulus encoding disappears during the SR, whereas choice-encoding remains of stable for both rules.

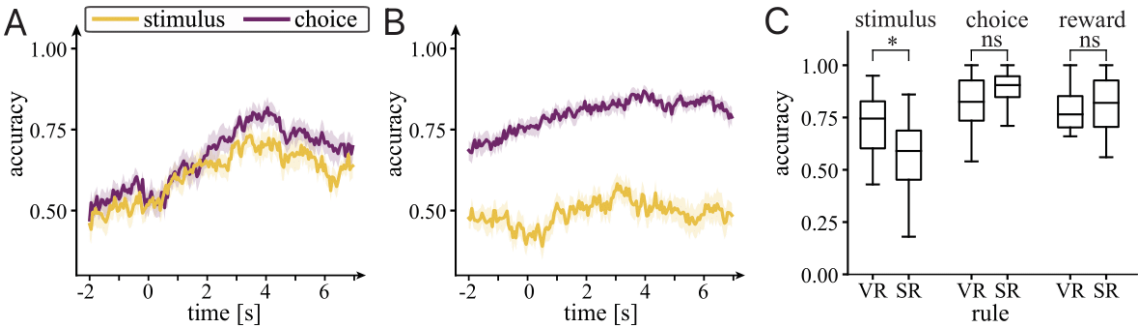

Figure 17: **Dynamic decoding probabilities of stimulus and choice. (A)** Decoding accuracy of stimulus and choice from neural recordings during VR performance. Aligned at cue onset (time 0). **(B)** Decoding accuracy of stimulus and choice from neural recordings during SR performance. Aligned at cue onset (time 0) as well. Cue stimulus is non-inferable from neural recorded data. **(C)** Comparison of decodability between VR and SR trial markers. Only for stimulus decoding a significant drop is observed during the SR ($p < 0.05$).

To establish a reference frame for comparing neural states across rules in terms of their choice encoding, I developed a robust choice decoding framework based on the selection of a subset of recorded units that provides the best decoding accuracy for both rules. An LDA (see Methods) was trained on neural states of the choice phases of VR trials during the stable performance period and tested on the stable performance period of the SR trials, ensuring robust decoding for both rules (Figure 18A).

Since decoding behavior from neural states of both rules with one decoder comes with specific challenges , a subset of units was selected through a stepwise elimination procedure that maximized decoding accuracy for both rules (Figure 18B). The subset of units achieving the highest decoding accuracy while preserving the largest number of units was selected. This subset consistently had a significantly better decoding accuracy than the full population, achieving almost perfect choice decoding for both

rules($p < 0.05$; Figure 18C).

The resulting robust choice decoder provides a consistent discrimination measure for neural states between choices for both rules, thereby making neural states comparable. Moreover, the robust choice decoder can also be used to decode behavior from model-generated neural trajectories, thereby providing numerous analysis possibilities on the basis of behavioral features.

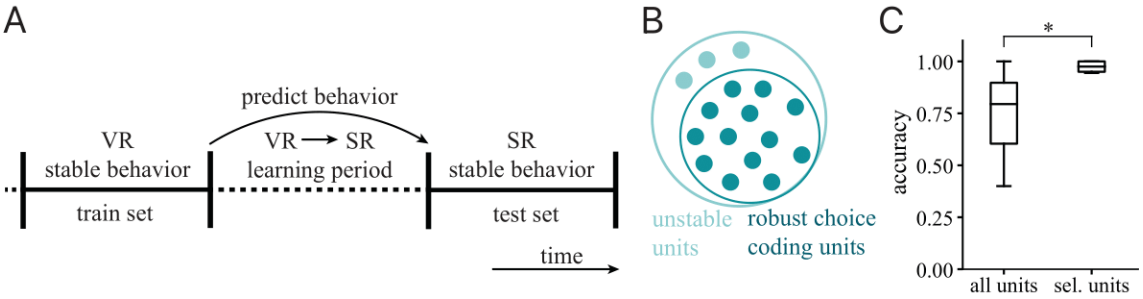

Figure 18: **Robust choice decoding framework. (A)** Illustration of the training concept. Trials from the stable performance period of the VR serve as training set for the linear classifier. Decoding accuracy of the classifier is evaluated on the stable performance period of the SR. **(B)** Illustration of unit subset. A subset of units in the full population changes their mean firing rate such that the stationarity assumption of the linear classifier is violated. Therefore, only the units that maintain consistent choice representations across both rules are selected. **(C)** Comparison of decoding accuracies between the full unit population and the selected subset. The selected subset is significantly better in decoding choices from both rules than the full population ($p < 0.05$).

3.2.2 PLRNN Reconstructions of Neural Activity

To assess whether the pePLRNN provided a valid reconstruction of the neural data, I defined a set of model evaluation criteria adapted to the specific challenges of experimental paradigm and the structure of the data. These challenges include the non-stationarity of the underlying system, noise inherent to neuronal activity, and the influence external inputs (like cue stimulus or decision phase). The model was evaluated by the following criteria:

1. accurate reconstruction of training data, reflected in correlation between model-generated trajectories and recorded activity,

2. preservation of task-relevant information, measured as decoding accuracy comparison for task variables between the reconstructed trajectories and the original data trajectories,

3. appropriate tracking of non-stationarity over time, assessed through agreement between inferred CPs from generated and recorded trajectories,

4. generalization to unseen data, evaluated via reconstruction performance on held-out trials,

5. physiological plausibility of long-term simulations, requiring the absence of unrealistic attractor such as isolated fixed-points in the autonomous regime.

These evaluation criteria serve both as internal validation of the model's fit and as basis for interpreting the reconstructed dynamics in the context of cognitive flexibility.

After training, the pePLRNN generated neural trajectories solely by providing the trial-individual initial condition as well as the trial-specific external inputs: cue presentation, choice period and reward delivery (see section 2.2).

The reconstructed activity closely followed the recorded neural signals, capturing both slow and faster time scales of units activity as well as input-driven and autonomous activity patterns (Figure 19A). Across all sessions, the pePLRNN generated neural trajectories that aligned well with the original recordings. Importantly, the reconstructed trajectories captured the dynamics of neural trajectories of both rule conditions, indicating that the model implemented the task-dependent shifts in neural activity.

To assess whether simulated trajectories and recorded neural activity exhibit the same decoding properties for task-relevant information, I used LDA (see Methods 2.5) to decode trial-specific features: cue site, choice, reward, and rule type. As shown in Figure 19C, there was no significant difference in decoding accuracy between the recorded and reconstructed population activity. Except for the cue site where the model performed better than the recorded data, likely because the site of the cue is specifically provided to the model as external input.

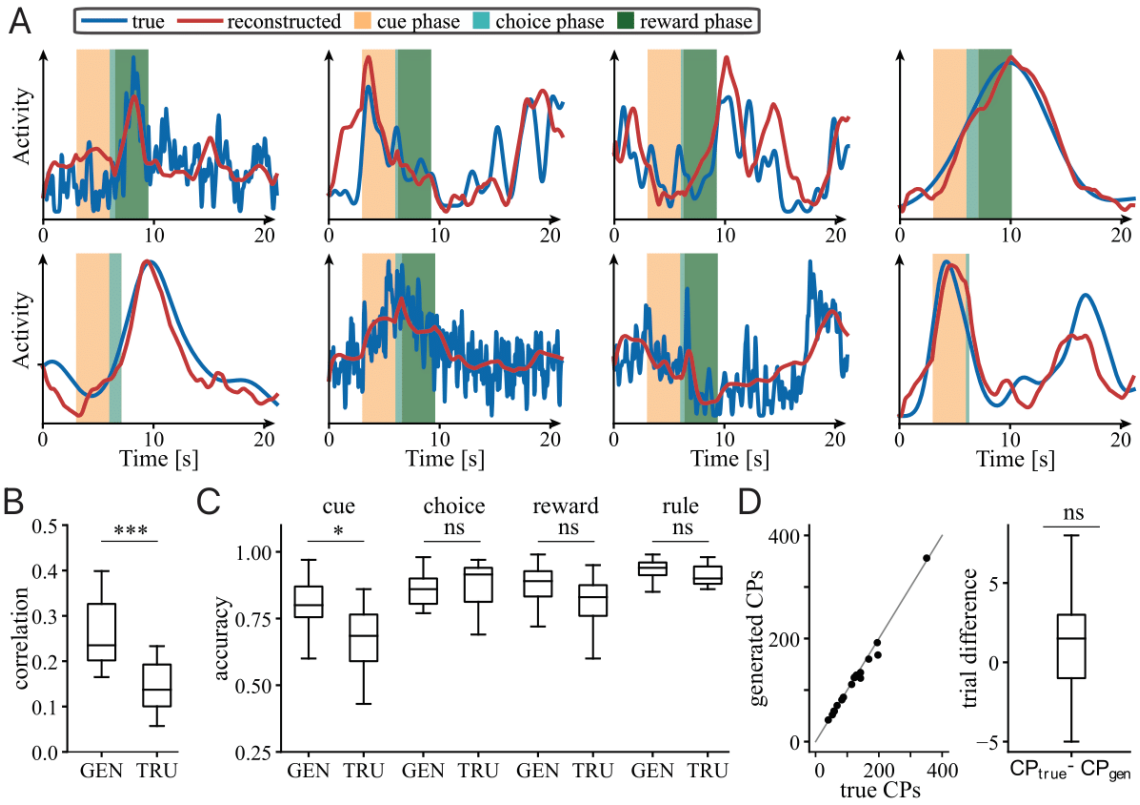

Figure 19: **DSR model accurately reconstructs neural activity and associated metrics. A** Examples of recorded (blue) and reconstructed (red) neural activity traces from test data. Showing that the pePLRNN captures both slow fluctuations and rapid transitions in the data. **B** Distribution of Pearson correlation coefficients between each reconstructed trace and its corresponding recorded trace versus the distribution of inter-trial correlations among recorded traces. Correlations for reconstructed-to-recorded pairs are significantly higher ($p < 0.001$). **C** Choice-decoding accuracy obtained from recorded and reconstructed population signals, showing no significant difference (mean ± SEM; n.s.,) for any category but cue site ($p < 0.05$). **D** Alignment of neural change-point times detected in reconstructed versus recorded data. Scatter of generated (gen) versus true (true) transition times falls along the unity line (dashed), with $R^2 = 0.98$, indicating near-perfect temporal agreement. Right panel: the boxplot shows that the mean of the distribution of trial-differences between recorded and generated neural CPs is not significantly different from zero (n.s).

Simulating Trial Transient

Before analyzing how the neural dynamics during the learning period change on a trial-by-trial basis, I first focused on the two behaviorally stable performance periods (see Methods 2.1 for details) to characterize how the neural dynamics of both rules are reconstructed by the model.

To validate the model's ability to reproduce rule-specific behavior during behaviorally stable performance periods, I used the trained pePLRNN as a surrogate model for the neural dynamics to generate distributions of neural trajectories under

experimental conditions (see Methods5 for exact details). Neural trajectory distributions were generated from rule-specific initial condition distributions for the four experimental conditions: VR with right-cue stimulus, VR with left-cue stimulus, SR with right-cue stimulus, and SR with left-cue stimulus (Figure 20A).

To obtain the behavioral readout associated with each generated trajectory I used the robust behavioral decoder (see Methods2.5) to transform generated trajectories into the decision discrimination space. Considering the final state of each trajectory as the choice-determining state, I obtain distributions of choice discrimination values associated with each experimental condition (Figure 20B). These distributions of choice discrimination values directly translate to behavioral choices, simply by their sign.

The model accurately replicated the desired behavioral patterns for both rules across all datasets. During the VR, the cue stimulus determined the choice outcome, producing a highly significant separation in choice discrimination values. In contrast, during the SR, the model-generated trajectories are decoded as a single choice output regardless of cue stimulus (Figure 20C). The behavioral distributions predicted by the model showed a high correlation with empirical behavioral data during stable rule periods (see Figure 20D, Spearman's $\rho = 0.79$).

Since the SR changes the reward contingencies of either site rather than directly specifying a left or right choice, I introduced two new categories to describe the two cue and choice alternatives: *the SR extinguished site SRES*, where responding is not reward (under the SR, and *the SR reinforced site (SRRS)*, which continues to be rewarded with same reward probability during the SR. Introducing these new categories allows me to standardize further analyzes and directly compare animals trained with a SR reinforcing left choices to those with a SR reinforcing right choices.

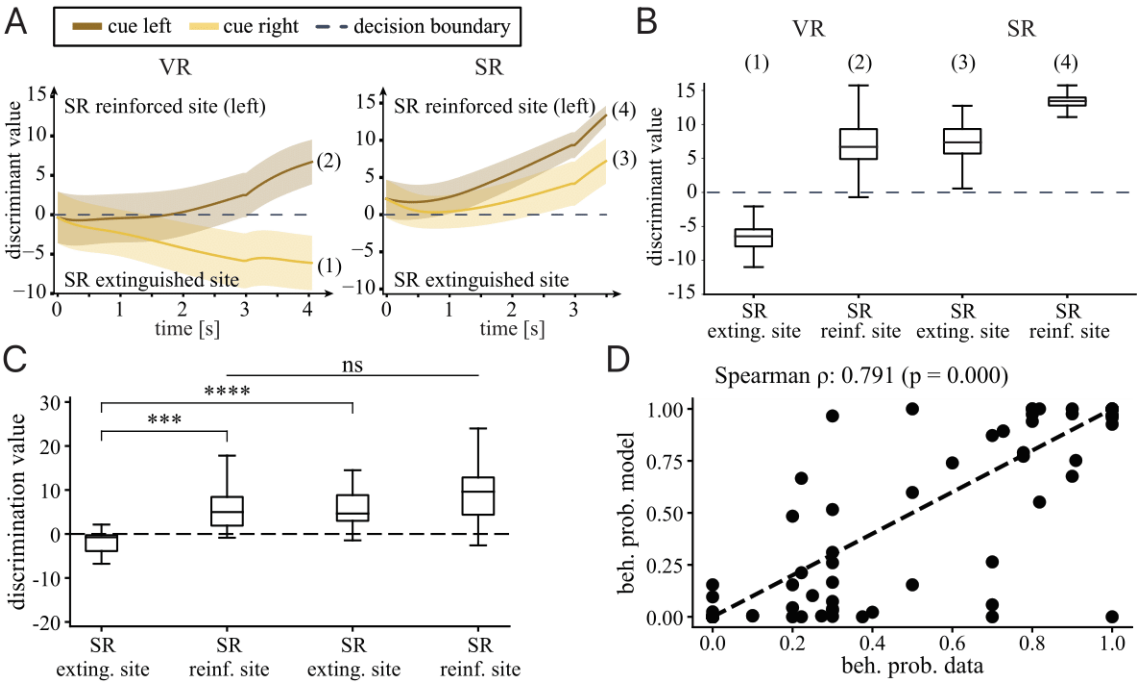

Figure 20: **Model-generated trial transients and recovery of behavioral statistics from random initial conditions. A** Representative trajectories simulated by the fitted pePLRNN in choice-discrimination value space, under spatial-reinforced (left) and spatial-extinguished (right) conditions during the visual-rule (VR) and spatial-rule (SR) epochs. **B** Distribution of simulated trajectory endpoints for the trials shown in A, demonstrating that the model settles into distinct decision locations corresponding to each rule condition. **C** Summary of transient discrimination-value distributions across all sessions. The shift of choice-discrimination value distribution of the SRES-condition from VR to SR is highly significant ($p < 0.0001$). This shift is larger than the initial difference between choice-discrimination value distributions SRES and SRRS during the visual rule ($p < 0.001$). **D** Comparison of behavioral choice probabilities during stable rule periods: empirical data versus model predictions, with Spearman's $\rho = 0.79$ ($p < 0.0001$), indicating that the model recovers the observed choice statistics.

In case the animal did not perform both rules correctly the model accurately capture this divergent behavior(see Figure21C)

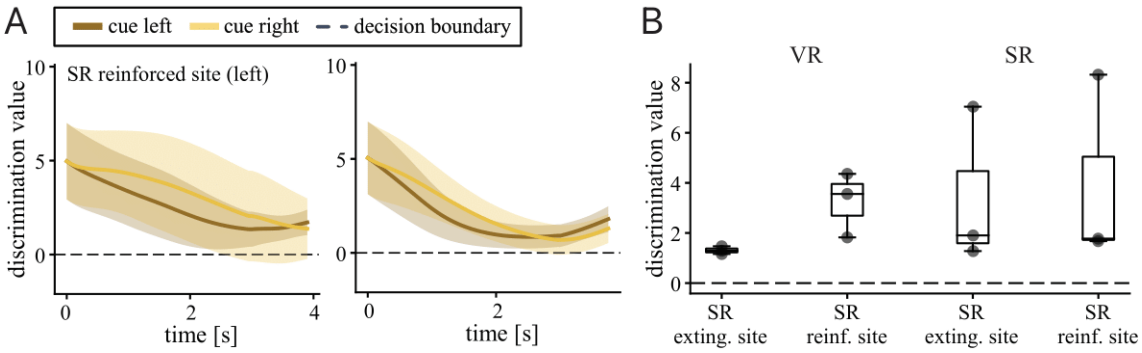

Figure 21: **The pePLRNN captures behavior of sessions, in which the animals performed only the SR A** Example discrimination-value trajectories from a session in which the animal performed only the SR. **B** Summary across all exclusively-SR sessions ($n = 3$): Both distributions lie significantly on the same side of the decision boundary.

3.2.3 Shifts in Stimulus-Dependent Attracting Regions as a Mechanism for Rule-Learning

The transient dynamics during each trial phase are influenced by input-dependent attractors. Similar to the case distinction made for the reconstruction of task-trained RNNs, I separately analyzed the two cue conditions of the two rules to identify case-specific attractors, with the same approach used in section3.1.1. Long-term simulations were used to identify sets of converged states that are linked to an attracting region (see Methods5). During the VR, trajectories of two different cue conditions converge toward distinct attractors with distinct behavioral decoding. For instance, right-cue stimuli led to trajectories converging in an attracting region decoded as a right choice, while left-cue stimuli trajectories converged in a left-choice-decoded attracting region (Figure 22A, left panel).

The transition from the VR to the SR results in a significant shift in the location of the cue-dependent attracting region associated with the SRES. The choice discrimination value of the SRES cue attracting region shifts significantly toward the attracting region associated with the SRRS. This effect is significant across all sessions($p < 0.05$; Figure 22C). In contrast, the cue-dependent attracting region associated with the SRRS site does not show significant shift compared between the visual and SR.

Despite the change in discrimination value, the cue-specific attracting regions for SRE and SRR remain distinct during the SR. They are located in separate regions in state space, as illustrated in the two-dimensional projection of converging trajectories (Figure 22D).

However, these case-specific attracting regions are never reached by the neural trajectory within the behavioral reaction times of the animals. These attracting regions primarily influence the transient trajectories rather than serving as endpoints during behaviorally relevant periods.

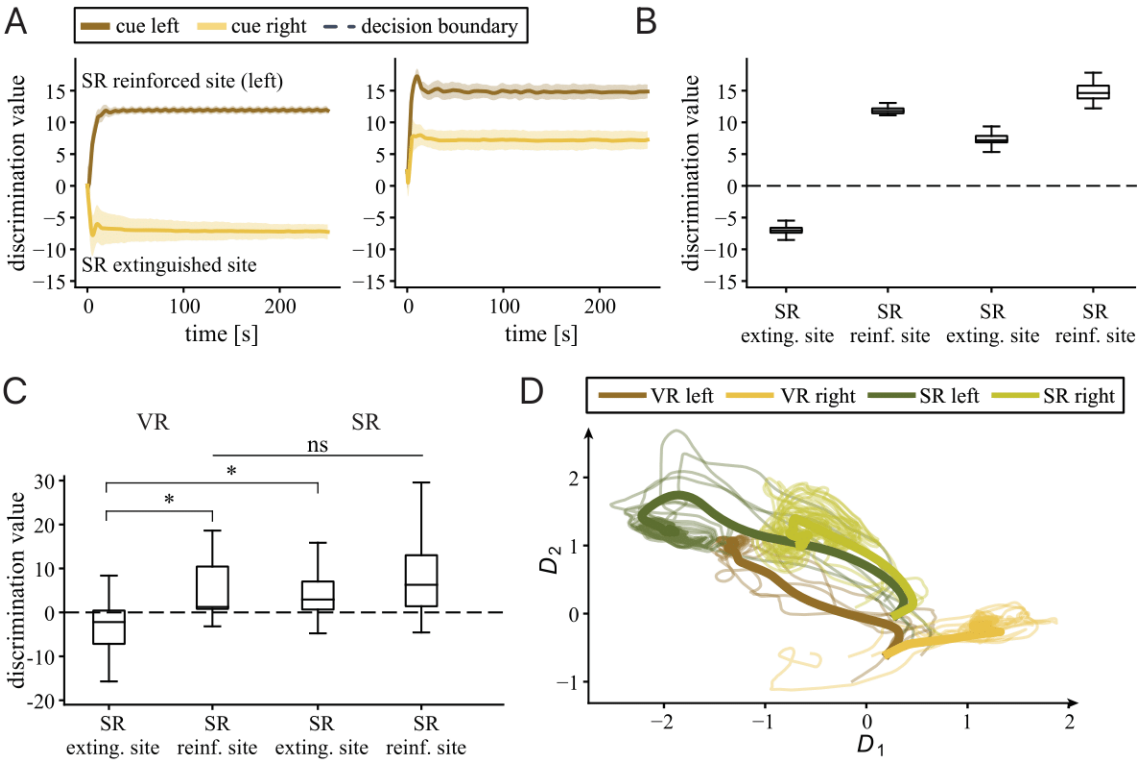

Figure 22: **Cue attracting region dynamics change across rules. A** Example discrimination-value trajectories evoked by left- (blue) and right- (red) cues during the visual-rule (VR) and spatial-rule (SR) epochs. The horizontal dashed line marks the decision boundary and shaded bars indicate cue presentation. **B** Boxplots of trajectory endpoints (attracting region locations) immediately after cue offset for the SR reinforced (left) and SR extinguished (right) sites. The SR-extinguished attracting region is systematically shifted toward the reinforced-site location. **C** Summary across all sessions showing the mean endpoint shift of the SR-extinguished attracting region relative to the reinforced-site location($p < 0.05$), demonstrating a significant cue-dependent remapping. **D** State-space projection of the four cue attracting regions—VR left, VR right, SR left, SR right—in the D_1–D_2 plane. Each mean trajectory (large marker) and its surrounding points indicate the robustness and separability of the cue-dependent attracting region states.

Change in the Neural Flow is the Main Cause for Change in Behavior

I examined how the two central components of neural dynamics, the initial condition and the parameters, each contribute to differences in the distributions of choice discrimination values, representing behavioral outcome distributions. Both components are rule-specific, potentially affecting the neural trajectory and the resulting behavior.

To distinguish the effects of initial conditions and the parameters, I performed a cross-condition simulation experiment. Trials were simulated for both stimulus conditions (reinforced or extinguished cue site during the SR), swapping initial conditions and dynamic parameters (Figure 23A). In the aligned condition, where both initial condition and dynamic parameters match the rule, the increase in choice

discrimination value (toward the SRRS) was asymmetric: the SRES cue site during the SR exhibited a significantly larger increase compared to the reinforced site (Figure 23B).

To isolate the role of the initial condition, I fixed the dynamic parameters and varied only the initial conditions across simulations. This had little to no effect on the choice discrimination value distribution across recorded sessions (Figure 23C). In contrast, there was a highly significant increase in the distribution of choice discrimination values when the initial condition was fixed and the parameters were exchanged. (Figure 23D). Again, the shift was significantly larger for the spatial-rule extinguished cue site compared to the reinforced cue site.

This together shows that the change in parameters is the main factor for the change in behavioral outcome.

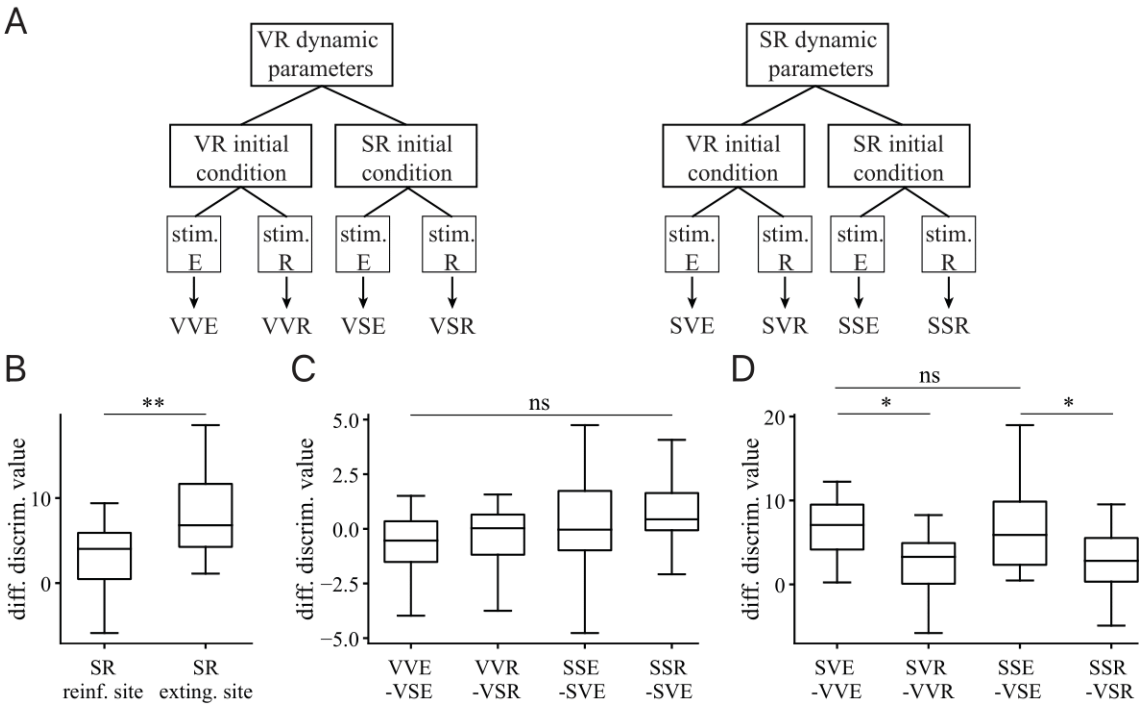

Figure 23: **Isolating the roles of initial condition and parameters in discrimination-value shifts.** **A** Schematic of the eight simulation conditions, combining the rule-specific (VR: V or SR: S), the rule-specific initial conditions (same as for the parameters: V or S), and stimulus conditions (reinforced: R or extinguished: E), labeled as VVE (to read as: VR parameters - VR initial condition - extinguished cue site), VVR, VSE, VSR, SVE, SVR, SSE, SSR. **B** Shifts in transient discrimination-value of the spatial-rule extinguished vs. the reinforced site when both parameters and initial condition are aligned: the shift of the extinguished-site is significantly larger than the shift of reinforced-site ($p < 0.01$). **C** Discrimination-value shifts for swapping the initial conditions (holding parameters fixed) show no significant differences from zero (n.s., one-sample t-test), across all sessions. **D** Across all simulations, the increase in discrimination value induced by the parameter change (with fixed initial condition) is significantly larger than zero ($p < 0.05$). The shift of the extinguished site is significantly larger compared to the reinforced site.

No Sign of Multistability

To investigate whether the transition between rules is implemented mainly as a shift of input-dependent attracting regions (an input-dependent monostable mechanism), or by multiple attracting regions in the autonomous regime(a multistable mechanism), I tested whether multiple attracting regions could be identified in the autonomous regime of stable performance periods of both rules. Therefore, I searched for task-relevant attracting region candidates for each rule separately, using long-term autonomous simulations to find distinct sets of convergence states as sign for distinct attracting regions. These simulations were initialized with either the endpoints of cue-driven transients or directly with states of the input-dependent

cue-attracting regions representing distributions of task-relevant states for each rule. For comparison, I generated long-term simulations initialized with the same initial conditions used for generating the cue-driven transient or the cue-attracting region coming from the resting state distribution. These simulations initialized with resting states show that there is at least one rule-specific attracting region, containing the neural states of periods in which the animal is not engaged in the task (see Figure25 and see Figure24). This resting state attracting region changes location in state space when the animal transitions from one rule to the other as shown in Figure 25C (cue-driven transients) and Figure 24C (as A1/A2). To find additional attracting region candidates, I used the endpoints of cue-driven transients as initial conditions. Regardless of the cue-driven initial condition, all trajectories of one rule converged to similar rule-specific sets of states (Figure 25A). Comparing the distributions of converged states of trajectories with cue-driven initial condition to the distribution of converged states of trajectories without cue-driven initial condition showed no significant difference in choice discrimination value within one rule (n.s.; Figure 25B). Moreover, comparing the same distributions for their difference in location in state space to the rule-specific resting state using the Wasserstein distance showed uniformly low to no difference in all cases. (Wilcoxon signed-rank, n.s.; Figure 25C).

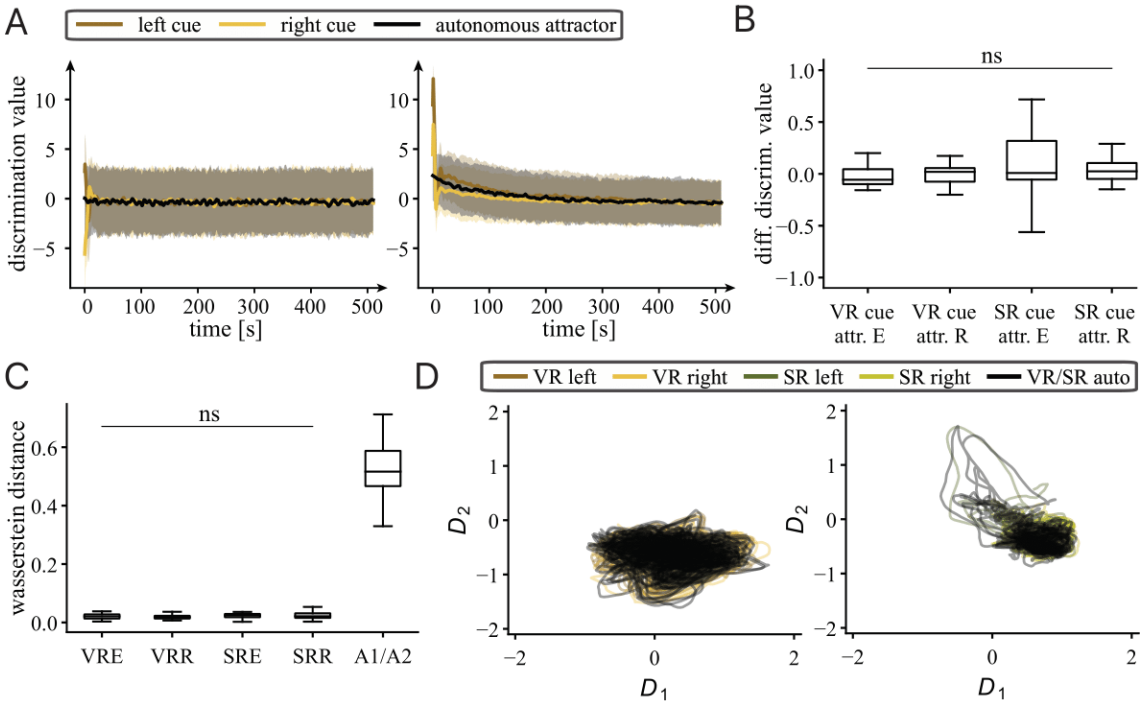

Figure 24: **Trajectories with input-driven initial condition converge to resting state attracting region. A** Example of trajectories in choice discrimination value space. Each initialized from one of the four cue-driven transient state distributions (VR-left, VR-right, SR-left, SR-right). All trajectories (colored) with cue-driven initial condition rapidly converge to the resting state attracting region (black). **B** Boxplots of the difference in discrimination value between the distribution of converged states and the distribution of resting states for each initialization for all sessions. Means of the distributions across conditions are not significantly different each other (n.s.). **C** Wasserstein distances between the distribution of converged states and the distribution of resting states for each initialization, showing uniformly low distances (n.s. for all comparisons among all possible pairs), confirming the similarity between sets of converged states. Resting state attracting regions of each rule are in different locations in state space. **D** Two-dimensional state-space projection of converged trajectories in the D_1–D_2 plane. The four trajectories with cue-driven initial condition (colored markers) overlap with the trajectories with resting state initial condition.

Second, to confirm that the input-dependent cue-attracting region states themselves do not converge to distinct sets of states, I repeated the long-term autonomous simulations with initial conditions directly from the cue-attracting regions. Again, all trajectories of one rule converged to similar rule-specific sets showing no significant differences in location or discrimination value compared to the respective rule-specif set of resting states (n.s.; Figure 24B-C). The overlap in state space between the trajectories with and without special initial conditions is illustrated as two-dimensional projections in Figure 25D (cue-driven transients) and Figure 24D) (cue-attracting region states).

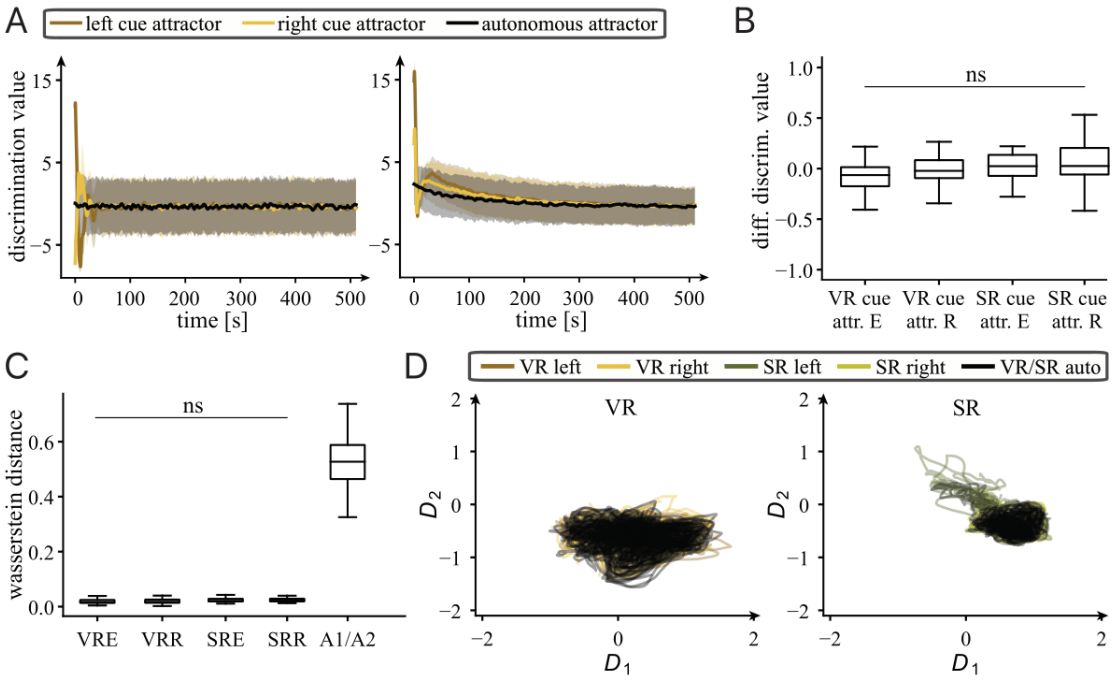

Figure 25: **Absence of multistability under autonomous dynamics—decision states are stimulus-driven. A** Example of discrimination value of trajectories of long-term simulations, each initialized from one of the four cue-attracting region distributions (VR-left, VR-right, SR-left, SR-right). Trajectories from all conditions converge to the resting state limit set (black). **B** Boxplots of the difference in discrimination value between the distribution of converged states and the distribution of resting states for each initialization for all sessions. Means of the distributions across conditions are not significantly different from zero (one-sample t-test, n.s.). **C** Wasserstein distances between the distribution of converged states and the distribution of resting states for each initialization, showing uniformly low distances (Wilcoxon signed-rank, n.s. among each other), confirming the similarity between sets of converged states. **D** Two-dimensional state-space projection of converged trajectories in the D_1–D_2 plane. The four trajectories from cue-attracting region initial condition (colored markers) overlap with the mean trajectories.

Together, these results demonstrate that, in the autonomous regime of each rule, all trajectories generated from task-relevant states converge to a single rule-specific resting state attracting region. Among the tested conditions, there is no evidence for multistability in autonomous regime for any of the two rules. Importantly, there remains a substantial separation in state space between the resting state attracting regions of the two rules across sessions and animals.

3.2.4 Trial-to-Trial Analysis Reveals Abrupt Transitions

After analyzing the dynamic features of the two rules during stable performance periods, I next investigate how model parameters and attracting region structure change on a trial-by-trial basis after the rule change. With this analysis I demonstrate the temporal structure underlying the transition between rules.

The core feature exploited here are the trial-specific connectivity matrices W_1 and W_2. Treating the trial-specific connectivity matrices as a time series allows tracking the evolution of weights during learning (Figure 26A). For each session, a significant change point in the composite connectivity parameters ($W^{(k)} = W_1^{(k)} W_2^{(k)}, k = 1, ..., K$) was detected, consistently preceding the behavioral change point (Figure 26D). In addition, I validated that trial-specific parameters do not encode trial-specific events such as cue site, choice, or reward, but mainly reflect the underlying rule identity (Figure 26C). Hence, trial-specific parameters are not overfitting single-trial trajectories but capture the more global structure of the session.

Complementing this, I tracked the evolution of cue-attracting region position in choice discrimination space (Figure 26B). CPs in the discrimination value of cue-attracting regions were also detected to be prior to the behavioral change point (Figure 26F). After the cue-attracting region change point, the mean choice discrimination value for trials associated with the SRES increased significantly (Figure 26H). Additionally, the location of the cue attracting region in discrimination space increased significantly after the cue-attracting region change point across all sessions; Figure 26I).

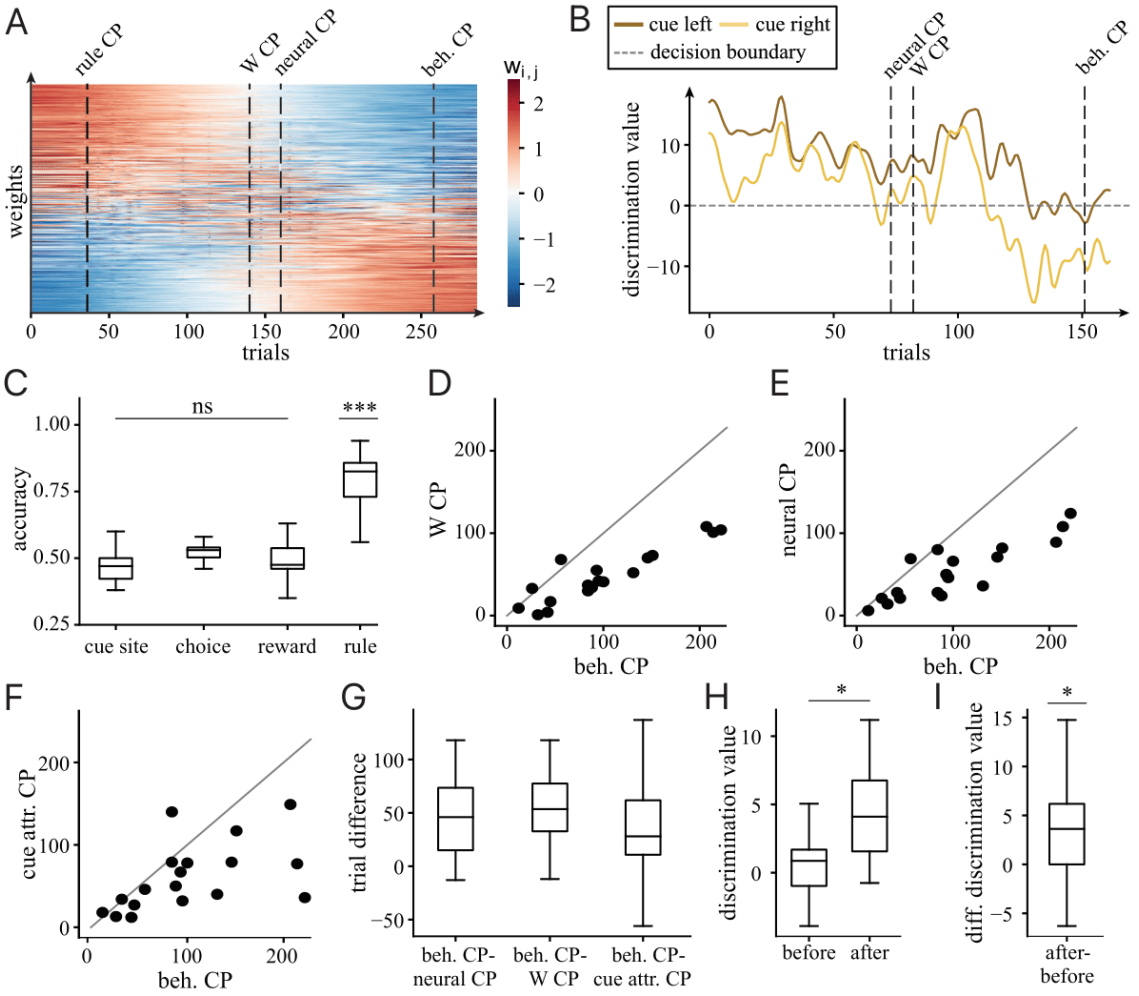

Figure 26: **Change points and alignment of model, neural, and behavioral change-points. A** Example of trial-by-trial evolution of connectivity matrix $W^k = W_1^k W_2^k$ (single weight evolutions, standardized and sorted by their linear slope). Vertical lines mark change-points: rule change (rule CP), parameter-change (W CP), neural-change (neural CP), and behavioral change (beh CP) point. **B** Example of trial-by-trial evolution of cue-attracting region location (in choice discrimination value space), with CP markers demonstrating temporal alignment of attracting region location change. **C** Decoding accuracy of $W^k, k = 1, ..., K$ for single-trial events (cue-site, choice, reward, rule); only the rules can be decoded above chance ($p < 0.001$). **D–F** Scatterplots of change-point relations (all aligned by the respective rule CP): (D) W CPs vs. beh CPs, (E) neural CPs vs. beh CPs, (F) cue-attracting region CPs vs. beh CPs **G** Boxplots of trial-lag distributions beh CP − {W CP, neural CP, cue CP} (median ± IQR). **H** Paired comparison of transient discrimination values averaged across ten trials before vs. after the cue-attracting region CP ($p < 0.05$). **I** Comparison of cue-attracting region location shift before vs. after cue-attracting region CP ($p < 0.05$).

Units Reorganize their Effective Connectivity after the Behavioral Change Point

Following the analysis of trial-by-trial changes in model parameters, cue-specific attracting region locations, and neural trajectories, I next investigate whether effective connectivity among units also changes during the transition between the two rules. Therefor, I developed a method to extract effective unit connectivity from trained models on a trial-by-trial basis.

Effective connectivity patterns were extracted by generating short trajectories from a distribution of trial-specific initial states. The product of Jacobians along each trajectory was computed to capture the cumulative local effective connectivity among units. Averaging across trajectories gave an estimate of the mean effective connectivity for each trial (see Sect. 2.5 for details). From each resulting connectivity matrix, the top 10% of the absolute weight values were retained to construct a graph of the most dominant effective unit connections.

I used the Jaccard similarity index of effective connectivity graphs between trials, to measure how effective connectivity changes between trials (see Figure 27A for examples). Effective connectivity patterns became significantly more similar across trials after the behavioral change point compared to before ($p < 0.001$, Wilcoxon signed-rank test; Figure 27B). Comparing the similarity after the behavioral change point to a shuffle distribution generated from random CPs confirmed that the behavioral change point indeed marked a distinct similarity cluster. This cluster was significantly more similar than any random subdivision ($p < 0.01$; Figure 27C). In contrast, no significant differences were found when comparing the similarity cluster of trials before the behavioral change point with trial after the behavioral change point to the shuffle distribution of random subdivisions (n.s.; Figure 27D).

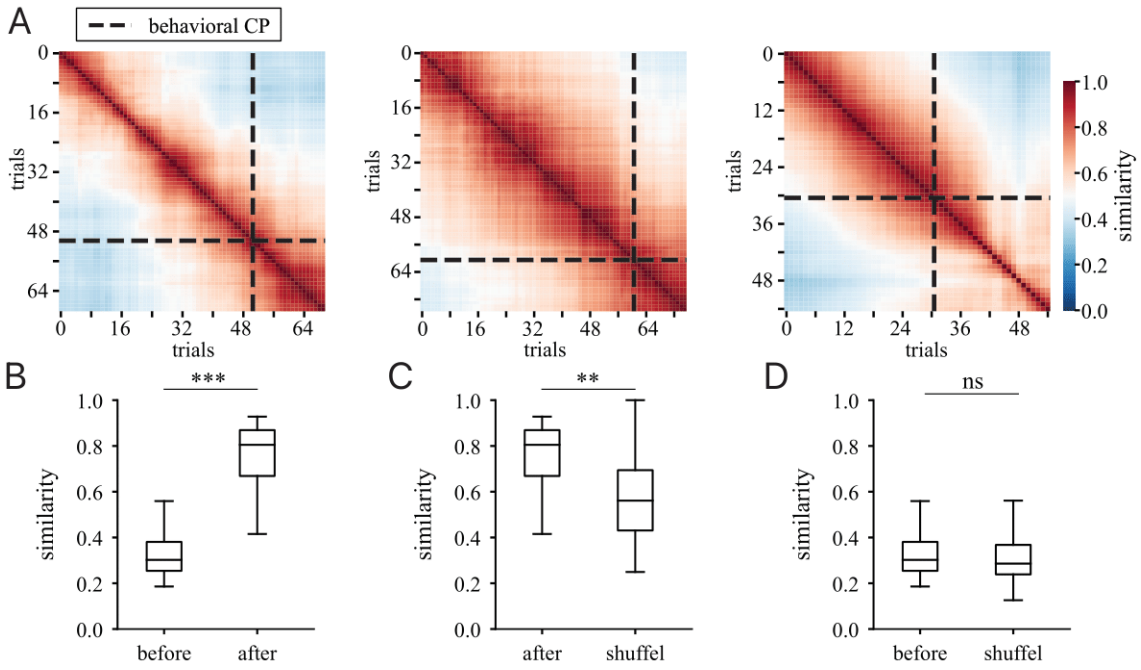

Figure 27: **Effective connectivity similarity increases after rule transition.** **A** Trial-wise similarity matrices of functional connectivity patterns derived from simulated trajectories for three sessions. Connectivity was estimated from trial simulations (specified in Section5). Each matrix shows pairwise Jaccard similarity of effective connectivity between trials. The behavioral change point separates these similarity matrices into clusters distinct clusters. **B** Mean functional connectivity similarity before and after the behavioral change point across all sessions. Similarity is significantly higher after the change point ($p < 0.001$). **C** Mean similarity after the behavioral change point compared to a shuffle distribution based on random CPs. Similarity after the behavioral change point is significantly higher than expected by chance ($p < 0.01$). **D** Mean similarity before the behavioral change point compared to the corresponding shuffle distribution. No significant difference is found (n.s.), suggesting that the increase in similarity is specific to the post-behavioral-change-point phase.

4 Discussion

The central aim of this PhD thesis was to uncover the dynamical computational mechanisms underlying rule learning by reconstructing the underlying non-autonomous DS with the pePLRNN directly from non-stationary data. Analyzing the trained pePLRNN as a functional surrogate model enabled the use of DST to describe how changes in behavior can be explained in terms of changes in temporal local vector field governing the neural state space. Given that rule learning in animals is an inherently non-autonomous and non-stationary process, approximating the dynamics through time-dependent parameters while also considering sensory stimuli provide the use of case distinctions to model parts of the time-dependent process as autonomous DS. Before reconstructing the neural data, I find that the pePLRNN is capable of reconstructing a broad spectrum of dynamical phenomena, ranging from fixed-point attractors, complex limit cycles to chaotic regimes in both discrete- and continuous-time systems. Multiple bifurcations within a single dataset can be accurately reconstructed. I use a series of methods to show that the pePLRNN can be used as functional surrogate model for extracting dynamic implementations of computational mechanisms in task-trained RNNs. Treating time-dependent parameter and external inputs as independent cases provides a twofold decomposition of non-autonomous dynamics into temporally local autonomous cases (*snapshot parameters*) and cases driven by external inputs. Each combination of these cases can be independently analyzed as autonomous DS in terms of their attractor structure. The model accurately reconstructs the non-stationarity of continuously learning task-trained RNNs, as shown by the agreement of stable attractor- and trajectory reconstructions across stationary and non-stationary periods. The pePLRNN reconstructed the neural dynamics underlying rule learning in the rat's mPFC on a trial-to-trial basis directly from data. To validate reconstructions, I define a set of criteria that ensure the model captures the task-relevant dynamic features, the non-stationary components and generates physiologically plausible long-term behavior. After training, the model generates trajectories that capture both the decoding characteristics and the non-stationarity of the original data. I used a robust decoding framework for decoding choices from neural trajectories across non-stationary rule conditions. I demonstrate that the pePLRNN can accurately recover the behavioral distribution of animals during stable performance periods under both rules using the robust choice decoder on model-generated trajectories. PePLRNN simulations for both task and rule conditions reveal that the main dynamic mechanism behind rule learning is a shift in the state space localization of stimulus-dependent attracting regions. This location shift of the attracting region was further validated as the primary mechanism, by showing that changes in the initial condition distribution have no significant impact on the resulting behavioral distribution. Furthermore, the reconstructed systems showed no sign for multistability, indicating that both rules are implemented as single attracting regions that depend on stimulus and rule context.

Trial-to-trial analyses of the reconstructed dynamics reveal that neural dynamics, snapshot parameters, attracting region locations, and behavior all undergo abrupt transitions during learning, rather than gradual adaptations. These transitions in

neural and dynamical variables consistently precede the behavioral change point. Using the behavioral change point as a reference, I demonstrate that the model-derived effective connectivity is organized into distinct similarity clusters, separated by the behavioral CP.

Using task-trained RNNs, I further show that the specific experimental structure influences the implementation of the computational mechanism used to encode the two rules. In the case of a continuous, low-noise stimulus until decision, the underlying mechanism is a single, stimulus dependent attracting region that shifts in location. In contrast, the same task structure incorporating a delay period between stimulus and choice is implemented as a multistable attracting region mechanism. This finding holds not only for the task-trained RNNs but also for pePLRNN-based reconstructions. Swapping input conditions, such that data from the continuous stimulus task are reconstructed assuming a delay period, yields a dynamical mechanism identical to the one inferred from the RNN trained on the delay task. Conversely, reconstructing the memory task under continuous cue input conditions leads to the single attracting region mechanism.

4.1 Advancing Dynamical Systems Reconstruction with pePLRNN

The pePLRNN introduced in this thesis provides an extension of DSR to the non-autonomous domain. Classical DSR methods typically rely on the assumption of time-invariance ([120, 209]), where the geometry of the vector field are governed by fixed set of parameters ([229, 120]). Such an assumption fails to hold the context of this experiment, where the underlying neural DS evolves in response to unexpected feedback, or in general biological systems which natural experience contextual modulation or drifts, and ongoing internal or external fluctuations (see Sect. **??**). The pePLRNN addresses this limitation by explicitly modeling time-dependent parameters, thereby enabling the reconstruction of non-stationary neural systems from empirical time series.

The central idea of the model is the formulation of non-autonomous dynamics as a sequence of temporally local autonomous systems. These local systems, *snapshot vector fields*, approximate the time-dependent vector field within specific time segments, regularized by a continuity prior to enforce smooth transitions between consecutive parameter regimes. This piecewise-smooth decomposition together with consideration with the external-input term allows for a precise dissection of non-autonomy into two distinct sources: The first is temporal changes in internal system parameters (e.g., reflecting plasticity), and (2) modulation by external inputs (e.g., stimulus-driven perturbations). The model thus enables a functional separation of the different sources of non-stationary into internally and externally induced transitions. This structure further allows for a hierarchical case distinction effectively constructing a tree of dynamical regimes, isolating the effects different sources of non-stationarity. Each separate regime can then be analyzed using standard tools from autonomous DST, compared to others to determine

Importantly, the pePLRNN can reconstruct latent dynamics across qualitatively different regimes, including systems undergoing bifurcations, oscillations, or chaotic

behavior. This would not be possible with a time-invariant parameter formulation, which would necessarily collapse qualitatively distinct transitions into a single averaged dynamical regime. The model thereby preserves topological and geometric properties of the system that would otherwise be lost with standard reconstruction techniques. The pePLRNN also functions as a generative model. After training, it allows for simulation of latent trajectories under varying initial conditions, external inputs, or parameters. These simulations can be used to extract distributions over behavioral outcomes, enabling direct comparison to empirical distributions. Moreover, the principle of using snapshot parameters (piecewise constant approximations to a time-dependent parameter trajectory) is not restricted to this model class. In fact, this approach can be applied to a wide range of recurrent neural network architectures. Moreover, the snapshot parameter principle is also not fixed to the internal parameters of system. It is also possible, for instance, to consider the parameters of an input-mapping function with snapshot parameters. As such, the snapshot parameter formulation represents a broadly applicable extension to existing models for analyzing non-autonomous systems Overall, the pePLRNN provides a flexible and interpretable framework for reconstructing and analyzing non-autonomous DS, by disentangle different sources of non-autonomy into separate autonomous DSs.

Role of Hyperparameters for Reconstruction The reconstruction performance of the pePLRNN is primarily regularized by the two hyperparameters: the global L2 penalty λ_1 and the continuity prior λ_2. While λ_1 enforces parameter sparsity by penalizing large magnitudes across all weights, λ_2 regulates the temporal smoothness of the evolving parameters, thereby controlling the model's capacity to capture non-autonomous transitions in system dynamics. This method approximating the time-dependent parameter trajectory resembles the principles of the Whittaker–Henderson smoothing for time series data (originally discovered by [20] and later rediscovered by [255] together with [103]).

A low value of λ_2 enables high flexibility in parameter variation across time segments, allowing the model to fit rapid dynamical changes. However, this increases the risk of overfitting, particularly under conditions of limited or noisy data. In contrast, large λ_2 values suppress temporal variability in parameters, effectively biasing the model toward stationary or mean-regime dynamics. This regularization strategy can obscure genuine bifurcations and distort latent trajectory structure if imposed too strongly. Additionally, λ_2 determines the temporal resolution of the snapshot vector fields. Smaller values lead to on over representation of single-trajectory features, while larger values smooth over dynamical details. The continuity prior thereby introduces a bias that implicitly defines the model's sensitivity to structural change. For instance, if λ_2 is too small, trial-dependent parameters will also adjust to trial-specific features like the stimulus or reward. Furthermore, if the continuity prior is too little, the model can be under-constrained with respect to the data leading to inconsistent state space geometry across consecutive temporal segments. Thereby artificial transitions in state space geometry might be introduced. The reason here is likely that the amount of data is too little for each parameter set to constrain the solution space enough. Conversely, a too large continuity prior could result in an underfitting of specific temporal-segments, like fitting only the neural

dynamics associated with the most abundant rule of the session. Importantly, continuity in parameter space does not imply continuity in vector field geometry: small parameter shifts near bifurcation points may still yield large qualitative changes in system behavior. Thus, λ_2 acts as a temporal filter on time-dependent parameters which promotes coherence in the geometry of consecutive reconstructed snapshot vector fields by constraining parameter evolution to a smooth path in parameter space.

Together, λ_1 and λ_2 define the expressivity and stability of the pePLRNN. Their careful tuning is essential to achieving reconstructions that are both dynamically rich and interpretable, while avoiding degenerate or over-constrained solutions.

Importantly, the continuity prior of the pePLRNN functions analogously to Whittaker–Henderson smoothing applied to the parameter trajectory. This form of smoothing corresponds to a zero-phase filter, meaning that it treats future and past events symmetrically (see Appendix5). As a result, when the continuity prior is active, abrupt transitions in the parameters are systematically broadened in time, resulting in reduced magnitude but temporally extended transitions that are symmetric around the transition point. In effect, this introduces a violation of causal structure, as future parameter values influence those assigned to earlier time points. However, this smoothing does not alter the actual location of the underlying change point in the trajectory.

4.2 Reconstruction of Ground Truth Data

Reconstruction of Benchmark Systems The reconstruction of controlled benchmark systems is an important evaluation of the pePLRNN's ability to reconstruct qualitatively distinct dynamical regimes from observational data. Using synthetic time series derived from the logistic map across a range of parameter values, I demonstrated that the pePLRNN can reconstruct a coherent series of vector fields that reproduce same bifurcation structure of the original system. This includes the recovery of fixed-points, periodic n-cycles, and chaos and their transition among each other. By reconstructing the different dynamic regimes of the DS underlying the bursting neuron model ([55]), I further show that the pePLRNN can reconstruct multiple complex limit-cycles and chaos from continous time systems with multipel time scales. These results confirm that the model is capable of representing DSs with both continuous and abrupt changes in structure, as required for modeling non-stationary phenomena.

Reconstruction accuracy depend on the regularization imposed by the continuity prior λ_2. Low values of λ_2 allow for more abrupt changes in parameter space, which led to overfitting and failure to recover consistent bifurcation structure. High values, in contrast, enforced smooth parameter evolution across time but constrained the system toward stationarity. This reduced the model's ability to reflect qualitative regime changes. Only within an intermediate range did the regularization allow sufficient flexibility for reconstructing transitions between dynamical regimes while maintaining coherence in the parameter evolution.

This balance reflects an inherent trade-off between model expressivity and temporal consistency. The continuity prior acts as a constraint that restricts the space

of admissible solutions to those with gradual parameter evolution. However, continuity in parameter space does not imply continuity in the vector field. Particularly near bifurcation points, small parameter changes may correspond to large structural differences in the dynamics. Therefore, even small violations of the continuity assumption can result in discontinuities at the level of latent trajectories.

The reconstructed bifurcation diagram of the logistic map illustrates that the model is capable of mapping the structure of a DS across different qualitative regimes, provided that regularization is appropriately tuned. Beyond the logistic map, further simulations with continuous-time systems confirmed the model's capacity to recover both simple and complex attracting region geometries, including cycles and chaotic trajectories. These results extend previous findings on stationary PLRNNs ([53, 127, 128, 210, 24, 104, 102, 23, 25, 27]) and show that time-varying parameter formulations allow the reconstruction of non-stationary systems with rich internal structure.

Together, these benchmark reconstructions demonstrate that the pePLRNN provides a tractable and in-principle usable approach for reconstructing DSs with non-stationarity from observational data alone. This establishes the model as a viable tool for the systematic recovery of computational dynamics in settings where ground truth mechanisms are either known or can be controlled analytically.

The Artificial Rule Learning Task The main objective of creating the animal rule learning task as artificial sequence-to-sequence paradigm described in Section 2.3 was to construct a fully controlled environment in which the DSR of a learning processe could be systematically tested. This approach enabled access to a known and analyzable ground truth system, serving two purposes. At first, as additional validation of the pePLRNN reconstruction abilities, and second, the analysis and exploration of candidate mechanisms underlying the behavioral requirements of the specific task structure (changing behavior over the whole session, implementing the correct decision policy on single-trial level). Three task variants were considered, all with certain specifications to test different factors. The first version had continuous stimulus inputs until the end of the decision period replicating the original experimental structure without additional constraints. The second version also had continuous stimulus inputs but required the RNN to produce a choice-specific output during the stimulus period, thereby enforcing an internal representation of the intended choice (referred to as "fixing"). The third version introduced a temporal delay between stimulus and choice periods, while maintaining the fixing constraint. With this version I wanted to test how the absence of constant choice predictor forces the network to stabilize internal. In other words requiring the RNN to be able to retain memory. All task variants incorporated randomized components (including noisy inputs, variable stimulus durations, and randomized choice timing) to prevent overfitting. The use of explicit output requirements during the stimulus and delay phases was inspired by the work of Rajalingham et al. ([187]) and served two goals: first, to examine whether enforcing such outputs already induces changes in internal dynamics (compared to no fixing requirement); and second, to support the training of RNNs on the tasks with memory demands.

Reconstruction of Task-Trained RNNs To investigate whether the pePLRNN can be used in principle to uncover a dynamic computational mechanism in a fully controlled environment, I reconstructed DS underlying the hidden state trajectories of RNNs trained on the artificial the rule-learning task that replicates the experimental structure without further constraints. The pePLRNN cloud be fully validated as surrogate model under these condtions. Reconstruction results, almost identical, to the task-trained RNN show that the computation underlying each single-rule behavior is implemented via input-driven dynamics. In this regime, each specific configuration of external inputs guides the hidden state trajectory toward a single attractor associated with the correct output indicating that when external information is continuously available, the system does not require internal memory traces or multistablity to implement the task. The continuous external input to RNN could be understood in two ways, first it acts as prefect predictor (or regressor). It provide the correct input to system in the moment where the output is required, hence the RNN can simply act feedforward network maping input to output. And secondly the constant input is also a constant forcing factor, hence requiring asymptotic stability of the system under all input conditions. The reconstructed snapshot vector fields from the pePLRNN revealed no qualitative changes in the internal geometry for all external-input and task conditions (no inputs-driven and no internal parameter-related bifurcation). This shows that the computational mechanism used by the task-trained RNN under continuous input conditions relies on exogenous signals to produce correct outputs, allowing a purely feedforward implementation of behavior.

Dissecting Computational Mechanisms from Task-Trained RNNs The pePLRNN framework enables to systematically isolate the contribution of DS variables, potentially contributing to the process of learning, into two distinct categories, namely changes in initial conditions and changes in system parameter. Since parameters of input-mapping, and the decoder are fixed during training.

To test the ability of the pePLRNN to uncover computational mechanisms, I used it to reconstruct the hidden state dynamics from vanilla RNNs trained on the artificial rule-learning task (see Sect. 2.3). Fitting the pePLRNN to the hidden state dynamics of task-trained RNNs enabled the reconstruction of snapshot vector fields across different learning stages, from VR, over the learning period to to SR. The pePLRNN successfully reconstructed distinct attractors associated with each rule and each external input condition using the case distinction as analysis tool. The attractor locations reconstructed by pePLRNN changed systematically during the learning period, aligning with behavioral transitions. Decoding the behavioral meaning of input-dependent attractors provided a principled mapping between task phase, vector field geometriy and behavioral output.

Importantly, the model also revealed that learning the SR be simple relocation of the stimulus-dependent attractors. In tasks driven by continuous input streams, the system remained monostable, relying on external signals rather than internal dynamics to implement behavior. In contrast, tasks with impulse-like input required sustained internal states and thereby induced bifurcations in the autonomous dynamics. These results clarify how task structure shapes the computational strategy deployed by the system and confirm that the pePLRNN can flexibly adapt to either

regime.

Finally, these reconstructions demonstrate that the pePLRNN can act as a functional surrogate for the original artificial agent, capable ofboth its latent computational architecture and its behavioral output distribution. This positions the pePLRNN not only as a tool for retrospective DSs reconstruction but also as a generative model capable of simulating the mechanisms underlying cognitive transitions.

4.3 Influence of Task Design on Dynamical Mechanism

Specific experimental conditions can have a large impact on the underlying dynamical mechanism. The major difference between the two artificial task structures (the one with the continuous input and the one with impulse input) the RNNs were trained on is that, while in the continuous-input case the system does not need a stable attractor representation of the stimulus, as the stimulus continuously acts as external driving forces providing stabilizing the system at the correct location in state space to produce the correct behavioral output. In contrast, in the case with impulse input, an internal attractor structure representing the two cue stimuli is needed, as the stimulus information needs to be stabilized over the delay period without external input in order to implement the task correctly. This shows that by introducing a memory requirement, the qualitative implementation of the dynamic mechanism changes completely. In the continuous input case, no bifurcation is needed to implement the transition, as a stimulus context is sufficient to implement both rules of the task (again as if the RNN acts as feedforward network). While in the memory case, there is a bifurcation during the leanring period toward the SR in the autonomous regime

4.4 Animal Behavior and Decoding

Animal behavior during the rule-learning experiment showed substantial variability in initial rule-bias and behavioral adaptation. While all animals showed at least once a successful rule switch, their individual behavior at the beginning of the experiment (the bias with which animals approached the first few trials of the session) was not uniform: some used the last rule of their previous session, while other simply sicked to the SR (as initial guess) once introduced and one animal consistently had the wrong rule bias, starting always with the first rule of its previous session. While the population of animals is too small to draw statistical conclusions about this effect, it provides a hint that animals might use different strategies to solve the task and develop a model about the meta-structure of the experiment, and that behavior of consecutive sessions might not be independent from each other. Further these initial rule biases could reflect internally maintained rule-specific representations that are stable across sessions. The task can thus be interpreted as probing not only learning during the session, but also strategy selection under uncertainty about the next presented rule.

Behavioral transitions were not gradual. Instead, animals showed abrupt shifts in behavioral performance. This finding further supports the long-lasting idea of

sudden behavioral shifts accompanied by sudden insights in the prefrontal cortex activity associated with rule switching [62, 9, 83].

Comparison between VR and SR performances showed that the SR was more persistent across sessions. This may reflect its structural simplicity: correct responses could be achieved by ignoring the cue in a substantial fraction of trials, reducing the need to maintain stimulus-specific mappings. Despite this asymmetry, animals never adopted globally maladaptive strategies, such as consistently choosing the opposite response. Behavior remained structured, rule-contingent, and sensitive to the current reward mapping.

Cue encoding was selectively reduced during SR execution. Neural representations of the cue stimulus were present during the VR but became indistinguishable across cue types during the SR. This suggests a rule-specific reconfiguration of stimulus representations in the underlying neural dynamics. The same cue inputs were either treated as behaviorally neutral or functionally collapsed into a common latent representation during the SR.

Construction and Application of the Robust Behavioral Decoder A particularly important result for analysis is that neural populations undergo significant shifts in firing rate during the learning. This change in mean firing rate presents a fundamental challenge for decoding behavior and other task-relevant variables. Non-stationarity violates the assumptions of many commonly used statistical models, including linear classification or linear regression, both of which require stationarity in the underlying distribution ([14]). As a consequence, the decoding properties of neural trajectories cannot be meaningfully compared across rules without accounting for this shift. To address this issue, I constructed a robust choice decoder. This decoder enables the comparison of neural states across both rules in terms of their choice decoding. The decoder is first trained to linearly separate neural states of the two different choices during the VR. This linear boundary is then used to select the units with stable choice decoding properties for both rule. This decoder provides a robust metric (the discrimination score) for comparing neural states for their choice decoding properties across rules.

There are several reasons why this approach was chosen. First, the periods of stable behavioral performance are relatively short, meaning that the available data for training a decoder under the assumption of stationarity is limited. This places the analysis near the sparse data regime. As a result, nonlinear models such as multilayer perceptrons are not applicable, as the number of available samples is insufficient to determine a stable decision boundary or a consistent discrimination metric. Second, training a decoder on the full population would explicitly violate the stationarity assumption, potentially corrupting the output of the model. Third, the structure of the task causes an additional constraints on decoder construction. Only the VR provides a valid train set to train a decoder for both choices, as during the SR only one choice is reinforced. This means that decoder training during the SR would require either the inclusion of incorrect responses or, in some cases, would not be possible at all due to the absence of responses to the SRES. With out the robust choice decoder, I would either use a decoder under violated assumptions, or if using two decoders for the separate rule have a complete under representation of

responses of the SRES.

4.5 Reconstruction of Neural Data

Validity Conditions and Model Properties Applying the pePLRNN to neural recordings requires careful consideration of when a reconstruction can be interpreted as valid. Unlike the synthetic case, the true underlying DS is unknown, and no ground truth trajectories (except the ones used in training) are available for comparison. For this purpose I introduced a list criteria (see Sect. 3.2.2) that ensures that model accurately captures the neural dynamics of rule-learning animals. As a brief reminder this list was: (i) reconstruction accuracy, (ii) preservation of task-relevant information, (iii) representation of non-stationarity, (iv) generalization to unseen data, and (v) plausibility of latent dynamics, excluding fixed point solutions. Each criterion fulfills a specific purpose. The first criterion simply tells us that the model can indeed fit the data. And thus is not fundamentally under-parameterized or over-regularized, this criterion does not provide any information about if the regularization might be too low. The second criterion is connected with the first one (as it requires a high correlation with the original data) and tells us in addition if model reconstructions encode task-information as good as the original data. The third criterion is used to make sure that the non-stationarity is correctly captured. If the smoothness constrain would be too high this would potentially result in a obscuration of the changes in neural dynamics. Thus this criterion can provide information if the smoothness constrain is too high. The fourth one asses that the model can generalize on periods of approximate parameter stationarity. This is necessary because classical ML validation techniques, such as train-test splits (cross-validation; [14]), cannot be used in the conventional way ([14]) in this context due to the inherent non-stationarity of neural data from learning animals. Since parameter evolution (i.e., synaptic plasticity [118]) is part of the learning process, temporal generalization becomes ill-defined. This prohibits the use of conventional test sets, as parameter values at time t do not predict those at time t', without any further assumptions. To address this, I identified behaviorally stable periods before the rule transitions where parameter changes can be reasonably assumed to be minimal. Within these segments, the assumption of approximate stationarity allows for evaluating reconstruction quality by comparing simulated and observed trajectories of held-out trials. However, this assumption is fragile as internal physiological processes such as heartbeat or breathing, fatigue, or strategy shifts may introduce latent non-stationarities, without directly impacting behavioral performance. Thus, any comparison must remain constrained to time intervals where behavioral output is consistent and accurate. The validation on unseen data provides information whether regularization in general is to low as a too low regularization will lead to over fitting.

Each criterion addresses a distinct aspect of model validation. The first criterion evaluates whether the model can accurately fit the observed data. This ensures that the model is neither fundamentally underparameterized nor excessively regularized. However, this measure does not indicate whether the regularization strength is too low, which could result in overfitting.

The second criterion is closely related, as it also requires high correlation between reconstructed and original data. In addition, it assesses whether task-relevant information encoding (such as stimulus, choice, or rule identity) is preserved in the model reconstructions. This ensures that the model does not merely reproduce surface-level firing patterns but also retains structured variability aligned with behaviorally meaningful dimensions.

The third criterion evaluates whether the model adequately captures the non-stationarity in the latent dynamics. If the smoothness constraint is set too high, it may suppress abrupt trial-wise changes and obscure dynamic transitions in the latent vector field. This criterion therefore provides insight into whether the continuity prior imposes excessive contains.

The fourth criterion test how well the reconstructed system generalizes to unseen data segments, under the important assumption of approximate parameter stationarity. Standard validation techniques (such as train-test splits) cannot be applied in this context [14] due to the inherent non-stationarity of neural activity in learning animals. Because parameter evolution (for example through synaptic plasticity [118]) is part of the learning process, temporal generalization becomes ill-defined. Parameter values at time t cannot be assumed to be the same at a later time t' without additional assumptions. To make validation possible, I used the behaviorally stable periods (see Sect. 2.1) preceding the rule change where parameter changes could be assumed minimal. Within these intervals, held-out trials were used to evaluate reconstruction quality by comparing observed and simulated trajectories in terms of their correlation. This assumption is fragile, as internal physiological processes (such as respiration, arousal, fatigue, or strategy shifts) may introduce latent non-stationarity that does not directly affect the observed behavior. Validation was therefore restricted to periods in which behavioral output was both stable and accurate. This criterion provides information whether the continuity prior is too little, since this will impair generalization by overfitting the training data.

The fifth criterion concerns the physiological plausibility of the reconstructed limit set structure. Specifically, reconstructions yielding fixed point attractors (during long-term simulations) were excluded. From a physiological perspective, fixed point attractors are unlikely to represent realistic neural population dynamics ([60]), as cortical systems exhibit continuous variability in firing rates ([192]), even if animals as not engaged in any task ([169]). Inherent neuronal noise ([46]) and physiological processes prevent convergence to a fixed-point equilibrium states under natural conditions (except death). Moreover, fixed point attractors fail to capture the temporal and geometric structure typically observed in neural trajectories ([60]). Unfortunately absence of a dedicated validation dataset for estimating long-term autonomous dynamics imposes additional constraints. Generating such a dataset would theoretically require isolating the animal from all structured sensory inputs during an awake state, without employing pharmacological or technical interventions that could alter neural dynamics. Even under such conditions, physiological fluctuations (i.e. breathing [[121]] heartbeat [[207]]) would likely induce non-stationary neural activity. Additionally the representational drift Consequently, validation approaches based on comparisons of limit set properties, such as state space distance or power spectra ([127, 24, 60]) for autonomous systems, are not applicable here.

Additionally, the interpretation of limit sets requires long simulation horizons well beyond individual trial durations, raising further questions about their functional relevance in this experimental context. Taken together, this justifies the exclusion of fixed-point solutions and supports the requirement that the reconstructed dynamics show a nontrivial long-term structure.

The pePLRNN trained on neural recordings satisfies all the above criteria (defined in section 3.2.2) necessary for interpreting it as a functional surrogate model. First, it accurately reproduces observed neural activity patterns, with high correlation between recorded and reconstructed signals. Second, the decoding properties of the reconstructed latent states are consistent with those of the original data. Third, CPs in the reconstructed parameters align with observed behavioral transitions, confirming the model's capacity to track non-stationarity. Fourth, the reconstructed dynamics generalize to short segments of unseen data, validating the model's internal structure under mild extrapolation. Lastly, the latent trajectories do not converge to fixed-points, which aligns with known biological constraints and supports the plausibility of the learned dynamics.

The model exhibits a further key property: it captures neural dynamics across different time scales, including slow drifts as well as rapid trial-specific transitions. This temporal flexibility enables the simulation of neural trajectories under varying conditions, allowing for many interventions and hypothesis testing.

The last step in model validation, transition almost to analysis is the generation of trial simulations that capture the behavioral distribution of the animal.

Reconstruction of Dynamic Mechanisms Underlying Rule Learning The reconstruction of dynamic computational mechanisms underlying rule learning is the central contribution of this work. To resolve this question, I subdivided the problem into five analytical components: (1) How are the two behavioral rules represented during stable performance periods? (2) What latent dynamic structures guide the neural trajectory toward rule-consistent decisions? (3) How do these structures differ between the two rules? (4) Could an alternative mechanism involving multistability explain the observed changes? (5) How do these dynamic changes evolve on a trial-by-trial basis during learning?

Across all animals, model reconstructions show a shared computational mechanism. During stable behavioral performance, both rules were implemented via stimulus-dependent attracting regions that modulated the transient neural trajectory. In the VR, the neural trajectory was directed toward distinct regions in state space depending on the stimulus identity, leading to different decisions. Under the SR, there are still stimulus-specific attracting regions, but both stimuli guided the trajectory toward the same decision.

During the transition period between rules, stimulus-specific attracting regions shifted their location in state space, and with them shifted the modulation of the neural trajectory. The shift, in terms of their behavioral decoding, was not symmetric across stimuli: the attracting region associated with the SRES underwent a significantly larger displacement than the region associated with the SRRS.

To test whether the rule-specific changes in behavioral distribution are influenced by the change in initial condition distributions (i.e., changes in resting states

across rules), I compared trajectory distributions generated with identical parameters but with rule-specific initial conditions. Simulating trajectories with different initial conditions had little to no effect. Instead, changes in the effective connectivity parameters were required to reproduce the observed rule-specific change trajectory distribution. Analysis of the autonomous regime (without any external inputs) revealed no evidence of multistability. Long-term simulations consistently converged to a single rule-specific attracting region, even when initialized from widely separated points in state space.Furthermore, although the position of this attracting region (in the autonomous regime) shifted during learning, this shift occurred along the decision boundary and did not alter behavioral outputs alone. These findings diverge from classical theories of decision making based on multistable attractor networks. The absence of multiple stable states indicates that rule representation, in this specific experiment, is not implemented through coexisting decision states in the autonomous regime. Instead, flexible behavior is achieved through context-dependent relocation of a single stimulus-dependent attracting region.

At first sight, these results might be interpreted as diverging from established theories of decision making [250, 249]. In the existing literature, decision making is often modeled using either drift diffusion processes or RNNs [249]. In these studies, RNNs implement perceptual decision-making through multistable attracting dynamics [249, 261]. Both approaches represent the idea of an integrator mechanisms that enables the accumulation of evidence form noisy signals toward a decision threshold. However, these models are typically based on classical perceptual decision making tasks, such as the random dot motion paradigm, in which subjects or agents must integrate noisy sensory inputs [249] over time to execute the correct decision. In other words, these models incorporate the conditions in which there is uncertainty over the actual stimulus. In contrast, the decision-making part during single trials of the rule-learning task (presented), does not involve this type of noisy stimulus integration. Each cue stimulus is unambiguous and fully presented to the animal. Therefore it is plausible that the brain implements a different computational strategy in this context, one that dynamically guides the neural trajectory toward the correct response without the need of further internal stabilization of the stimulus information. In addition, the attracting regions determined by long-term simulations are in general not reached during real trials, simply because behavioral decisions occur substantially earlier. This observation further supports the idea that the modulation of the transient trajectory is more relevant to decision-making than convergence into a attracting region ([57]).

Limit Set Analysis Although an analytical method exists for identifying fixed-points and periodic limit sets for the clipped shallow variant of the PLRNN (SCYFI, [63]), this approach did not converge when applied to the trained models. Possible reasons why SCYFI did not converge are either the high-dimensional hidden space of the model (effectively leading to a tool large search space, see Sect. 2.2 and section5) or the possible presence of chaotic dynamics, which would be not detected by SCYFI. Due to these limitations, attracting regions were identified using long-term simulations. These simulations extended approximately ten times beyond the duration of an average trial, allowing the system to settle into stable temporal

patterns.

As a result, the results from these long-term simulations have to be interpreted as empirical sets or sets in attracting regions, and cannot be interpreted as rigorously defined attractors. Although the formal status of these limit sets cannot be rigorously confirmed, due to high hidden-dimensional settings required to produce physiological plausible long-term generations, they nevertheless provide meaningful insights into the system's long-term behavior. They provide a practical approximation to the potentially underlying attractor structure and allow for a qualitative assessment of convergence patterns and their functional roles in state space.

Trial-wise Transition Dynamics and Change Point Structure Trial-by-trial analyses of parameter evolution, location of attracting region in decision space, neural activity, and behavioral output showed significant CPs after the rule change. These CPs occurred abruptly rather than gradually. Changes in parameters and neural firing patterns systematically preceded the behavioral transition. These results support the hypothesis that rule learning is accompanied by sudden transitions in neural states and moments of sudden insight rather than slow adaptation ([62, 122]). The fact that CPs in parameters, attractor location and neuronal activity precede changes in behavior aligns with the hypothesis that mPFC undergoes representational change before behavioral adaptation ([183]). These early change points in PL might be interpreted as sign of suspecting change in environmental condition opening starting a period of high behavioral variability and flexibility.

The Computational Mechanisms of Task-Trained RNNs vs. Neural Recordings Comparing the reconstructed mechanisms from task-trained RNNs and neural recordings reveals both structural similarities and fundamental differences in coding. In both systems, behavior under each rule was implemented through input-driven attractor dynamics. However, the task-trained RNN employed simple fixed point attractors that acted as strong stabilizing anchors, with rapid convergence ensuring that each trial phase was represented by a distinct attractor state. In contrast, the neural data exhibited no such convergence. The attractors inferred from recordings were not fixed-points but high-dimensional limit sets, possibly chaotic or hyperchaotic, and trajectories never settled into a single state within a trial. This indicates a fundamental divergence: whereas the RNN's attractors served both as transient guides and stabilizing endpoints, the attractor-like structures in neural data modulated trajectory direction without enforcing convergence. This raises the question of whether specific attractor types (like fixed-points, cycles, or chaos) carry computational significance beyond defining the regions of state space they direct activity toward. Furthermore, while the RNN showed a gradual learning trajectory with smooth transitions in attractor geometry, behavioral learning in animals coincided with abrupt attractor shifts ([62, 122]). This supports the view that the brain relies on qualitatively different mechanisms than gradient-based optimization, further questioning the functional role of classical attractor structures in neural computation.

Connectivity Analysis In addition to state space analysis, the pePLRNN can also provide structural properties such as an estimate of the effective connectivity from the reconstructed system. By simulating neural trajectories with a trianed model, one can compute the cumulative product of Jacobians along each trajectory, yielding a trajectory-dependent distribution of Jacobians and thereby providing an estimate of the effective connectivity distribution underlying the simulated dynamics (see Methods 2.5). Applying a similarity analysis to extract graph structure, I found that the behavioral change point significantly separates two distinct clusters of enhanced similarity across trials. This, in turn, may indicate that the emergence of similarity clusters reflects the full implementation of the newly learned rule. Even though other dynamic features such as parameters, attracting regions, and neural firing rates may have already transitioned past their respective CPs, further reorganization appears to occur that is not readily observable through behavior or other model-derived features. Analyzing these distributions further can reveal which units form functional assemblies under specific conditions. Or these matrix-valued distributions can be analyzed in terms of simple distributional properties, like mean and variance. For instance, if the variance of the effective connectivity distribution is small, then all trajectories, irrespective of their initial condition, pass trough to the same linear subregions in states space. In addition they could provide information about network properties (e.g. sparse vs broadly connected) or graph-theoretic features, such as the emergence of clusters or characteristic activity patterns under varying conditions. Additional properties of the system itself may also be derived.

4.6 Limitations

The pePLRNN framework provides a flexible architecture for reconstructing latent dynamics from time-series data. However, its expressivity can also introduces biases linked to model assumptions and input design. These biases can make mechanistic interpretations difficult, especially when external inputs and regularization are improperly adjusted. Such issues manifest through three primary mechanisms.

Continuity Regularization Bias The continuity prior λ_2, designed to regularize parameter evolution, plays a dual role in controlling expressivity and introducing bias. While it reduces overfitting by coupling parameter estimates across time, strong regularization can suppress genuine dynamical transitions. At the same time, weak regularization allows the model to overfit input-driven variability, conflating external modulations with intrinsic state changes. Hence, parameter continuity imposes structural assumptions that shape the inferred geometry of the latent vector field, especially in data regimes with limited signal-to-noise ratio. Importantly, continuity in parameter space does not guarantee continuity in phase space due to the nonlinearity of the vector field mapping. Small parameter perturbations near bifurcation points may induce large qualitative shifts in system dynamics, obscuring true transitions. Thus, regularization may enforce smooth parameter evolution while simultaneously distorting the underlying flow, introducing ambiguity in mechanistic interpretation.

Representation vs. Mechanism Trade-Off A deeper implication of these biases is the trade-off between representation accuracy and mechanistic faithfulness. When task structure is offloaded onto input design or continuity priors, the model risks reconstructing observed data without capturing the system's true computational mechanism. This can obscure multistability, bifurcations, or other critical dynamical features, leading to the false impression of monostability or feedforward behavior. Without explicit constraints on the hypothesis space, the model may converge on degenerate solutions that prioritize smoothness and input-driven mappings over genuine latent dynamics. This is especially critical when mechanistic inference is the primary objective, as it challenges the validity of the inferred trajectories.

Data Availability and Segmentation Constraints Data availability and segmentation choices introduce further constraints. Assigning too much data to a single parameter set underfits local transitions; too little leads to noisy and unstable estimates. Sparse data settings exacerbate this problem, limiting expressivity and convergence. Moreover, continuity priors impose implicit assumptions about the time scale of learning. If these assumptions mismatch true learning dynamics, reconstructions may miss critical transitions or overfit transient fluctuations.

Biological Realism and Validation Applying the model to real neural data poses additional challenges. Ground truth dynamics are unknown, rendering direct validation infeasible. Standard train/test splits are problematic due to the inherent non-stationarity of behaviorally recorded data. Approximate stationarity assumptions must be made, typically by identifying behaviorally stable periods. Yet these assumptions are vulnerable to internal factors such as fatigue, motivation, or spontaneous strategy changes. Furthermore, the model cannot generalize to future time points, as it lacks a parameterized formulation of parameter evolution.

Limitations also arise in the identification of limit sets. Analytical methods such as SCYFI failed due to high latent dimensionality or chaotic dynamics, necessitating long simulations to infer asymptotic behavior. These limit sets, while informative, are not analytically validated attractors, constraining their interpretability. Additionally, such simulations are computationally demanding and scale poorly to longer recordings.

Identifiability and Mechanistic Faithfulness A critical unresolved issue in the pePLRNN framework is identifiability. The flexibility of the model allows multiple parameter configurations to produce similar trajectories, raising concerns about whether inferred mechanisms reflect genuine system dynamics or optimization artifacts. Current implementations lack formal identifiability guarantees or theoretical bounds on reconstruction uncertainty, limiting confidence in mechanistic interpretations. Future work should prioritize theoretical analyses to constrain model solutions and introduce data-driven model selection strategies to penalize degenerate mappings.

Influence of Input Structure on Reconstruction The design of the input matrix has a mechanistic impact on the reconstructed DS. When constant input vec-

tors, instead of impulse vectors are used to reconstruct the hidden state dynamics of RNNs trained on the impulse-based rule-learning task, then fro are used to encode task contingencies, the model may not learn more complex autonomous dynamics by assigning task information directly from external inputs. This creates degenerate reconstruction in which the model reproduces observed trajectories through input-driven mappings rather than uncovering the true latent dynamics. As a result, the reconstruction may appear accurate but fails to reflect the system's internal mechanism. This issue is particularly pronounced when the input matrix provides more information to the system than the original system has (e.g., the continuous-cue stimulus for a memory-requiring task). This reduces task-specific constraints from the internal dynamics. In such cases, the pePLRNN may exploit the increased degrees of freedom to optimize a reconstruction that relies on unrealistic assumptions, effectively bypassing attractor formation. This produces a "mirage" reconstruction: the model appears to fit the data, yet the inferred mechanism does not correspond to the one implemented by the biological system. This might introduce a simplicity bias, favoring direct mappings from input to output over internally sustained computations, even when the latter may have been necessary in the original system. These findings highlight a critical methodological point: experimental assumptions and input encoding can fundamentally alter the reconstructed mechanism. If too much structural information is embedded in the inputs, the reconstruction loses its diagnostic power to detect latent dynamics. Therefore, input design must be treated as part of the modeling hypothesis. Reconstructions should be evaluated not only by their fit to the data but also by whether the underlying architecture imposes unjustified shortcuts that bias the interpretation of the latent system.

4.7 Outlook

The possibility of reconstructing non-autonomous DS with the pePLRNN opens the door to several new experimental and theoretical questions. The results presented in this thesis point directly to new experiments in which the mechanism discovered here could be experimentally tested. In this follow-up experiment, animals would be required to switch flexibly between two rules, where one rule actually explicitly requires the involvement of working memory while during the other rule the animal can solely rely on a cue stimulus. This could lead to questions about the mechanism by which the brain actually recruits working memory when needed and how learning that information that needs to be kept in working memory is actually implemented.

A direct and important methodological improvement of the pePLRNN would be an additional model that captures the temporal evolution of snapshot parameters. This would substantially improve the testability, interpretability, and applicability of the pePLRNN. Connected to this is the integration of non-autonomous DSR with control theory to improve the understanding of manipulations and their effect on neural systems. For instance, closed-loop paradigms, which adjust to the non-stationary nature of human physiology (e.g. incorporating a model of the neural drift), could lead to improved applications in brain-machine interfaces or deep brain stimulation.

From a systems neuroscience perspective, the pePLRNN framework could be extended to large-scale recordings across brain regions. As rule learning and behavioral adaptation are complex cognitive functions that involve a distributed network with the thalamus, striatum, and sensory cortices. Introducing specifically structured observation models or regularization techniques that reflect biological constraints could further enhance the interpretability and plausibility of non-autonomous DSR in neuroscience ([60]). This would move the analysis beyond local reconstructions toward a system-wide understanding of cognitive flexibility.

Many of the limitations of current pePLRNN framework can be solved by integrating multimodality in the current non-autonomous DSR framework. A highly promising work by Brenner et al. (2023) ([25]) already provides such an integration. Another perspective brings the idea of hierarchical models developed by Brenner et al. (2025) ([26]) originally developed to perform DSR within a group of different datasets to integrate them into a common model. This can be easily extended to the temporal segmentation used here in this to infer group level parameters with a specific time constraint. Both together would result in a powerful framework accounting for not only non-autonomy but also multimodal data, like behavior.

5 Conclusion

The main goal of this PhD thesis was to uncover the dynamical computational mechanisms underlying rule learning by reconstructing the underlying non-autonomous DS with the pePLRNN directly from neural recordings. Analyzing the trained pePLRNN as a functional surrogate model enabled the use of DST to describe how changes in behavior can be explained in terms of changes in temporal local snapshot vector field, governing the neural state space. Given that rule learning in animals is an inherently non-autonomous and non-stationary process, approximating the dynamics through time-dependent parameters while also considering sensory stimuli provided the use of case distinctions to model parts of the time-dependent process as autonomous DS. This idea of dividing non-autonomy first into two conceptual parts (input-driven and time-driven non-autonomy) and structuring the model such that it reflects this division provided as useful simplification of the underlying process. This was especially possible because both processes (the synaptical changes and sensory stimulation) work on different time scales. Before reconstructing the neural data, I demonstrated that the pePLRNN is capable of reconstructing a broad spectrum of dynamical phenomena, ranging from fixed-point attractors and complex limit cycles to chaotic regimes in both discrete- and continuous-time systems. Multiple bifurcations within a single dataset can be accurately reconstructed. Moreover, I developed a series of methods that use the pePLRNN as a functional surrogate model for extracting dynamic implementations of computational mechanisms in task-trained RNNs. Treating time-dependent parameters and external inputs as independent cases provides a twofold decomposition of non-autonomous dynamics into temporally local autonomous cases (snapshot parameters) and cases driven by external inputs. Each combination of these cases can be independently analyzed as autonomous DS in terms of their attractor structure. The model accurately reconstructs the non-stationarity of continuously learning task-trained RNNs, as shown by the agreement of stable attractor- and trajectory reconstructions across stationary and non-stationary periods. The pePLRNN reconstructed the neural dynamics underlying rule learning in the rat's mPFC on a trial-to-trial basis directly from data. To validate reconstructions, I defined a set of criteria that ensure the model captures the task-relevant dynamic features, the non-stationary components, and generates physiologically plausible long-term behavior. Once trained, the model generates trajectories that replicate both the decoding characteristics and the non-stationarity of the original data. I developed a decoding framework for robust decoding of choices from neural trajectories across non-stationary rule conditions. Using this robust choice decoder applied to the generated trajectories, I demonstrated that the pePLRNN can accurately recover the behavioral distribution of animals during stable performance periods under both rules. Reconstructions across task and rule conditions reveal that the main dynamic mechanism behind rule learning is a shift in the state space localization of the stimulus-dependent attracting region. This location shift constitutes the primary mechanism, since changes in the initial condition distribution have no significant impact on the resulting behavioral distribution. Furthermore, the reconstructed systems showed no sign of multistability, indicating that both rules are implemented as single attracting regions that depend on stimulus

and rule context. Trial-to-trial analyses of the reconstructed dynamics reveal that neural dynamics, snapshot parameters, attracting region locations, and behavior all undergo abrupt transitions during learning, rather than gradual adaptations. These transitions in neural and dynamical variables consistently precede the behavioral change point. Using the behavioral change point as a reference, I demonstrated that the model-derived effective connectivity is organized into distinct similarity clusters, separated by the behavioral change point. Using task-trained RNNs, I further showed that the specific experimental structure influences the implementation of the computational mechanism used to encode the two rules. In the case of a continuous, low-noise stimulus until decision, the underlying mechanism is a single, stimulus-dependent attracting region that shifts in location. In contrast, the same task structure incorporating a delay period between stimulus and choice is implemented as a multistable attracting region mechanism. This finding holds not only for the task-trained RNNs but also for pePLRNN-based reconstructions. Swapping input conditions, such that data from the continuous stimulus task are reconstructed assuming a delay period, yields a dynamical mechanism identical to the one inferred from the RNN trained on the delay task. Conversely, reconstructing the memory task under continuous cue input conditions leads to the single attracting region mechanism.

To close, the pePLRNN provides a usefull method to expand classical autonomous DSR to non-autonomous DSR while preserving all handy tools that autonomous DST provides. It is capable of reconstructing complex phenomena starting from fixed-points and ending at neural dynamics during rule learning recorded from performing rats. With its special structural properties I was able to discover a dynamic computational mechanism that explained and observed behavior in animals. It could also generate new hypothesis about the use of attractors as internal representations and the possible reliance of the brain on external stimuli. All this and all of the above, makes this work a valuable contribution on the path to resolving *the question* in neuroscience: *"How does the human brain work?"*. Pushing the limits of what can be said.

Appendix

Specific Experimental Protocols

Generation of the Bursting Neuron Benchmark System

The bursting neuron model from ([56]) was simulated to generate benchmark data for model reconstruction. The system consists of one voltage variable and two auxiliary gating variables governed by the following set of ordinary differential equations:

$$\frac{dV}{dt} = \frac{1}{C}\Big[I - g_L(V - E_L) - g_{Na}m_\infty(V)(V - E_{Na}) - g_K n(V - E_K) - g_M h(V - E_K) - g_{NMDA}s_\infty(V)V\Big], \tag{25}$$

$$\frac{dn}{dt} = \frac{n_\infty(V) - n}{\tau_K}, \tag{26}$$

$$\frac{dh}{dt} = \frac{h_\infty(V) - h}{\tau_M}, \tag{27}$$

with sigmoidal steady-state activation functions defined as:

$$m_\infty(V) = \frac{1}{1 + \exp\left(\frac{V_{h,Na} - V}{k_{Na}}\right)}, \quad n_\infty(V) = \frac{1}{1 + \exp\left(\frac{V_{h,K} - V}{k_K}\right)}, \tag{28}$$

$$h_\infty(V) = \frac{1}{1 + \exp\left(\frac{V_{h,M} - V}{k_M}\right)}, \quad s_\infty(V) = \frac{1}{1 + 0.33\exp(-0.0625V)}. \tag{29}$$

Integration was performed using the `solve_ivp` function from SciPy with a relative tolerance of 10^{-6} and absolute tolerance of 10^{-7}. The system was initialized with $V_0 = -60$, $n_0 = 0.0$, and $h_0 = 0.01$, followed by a 100-step initialization phase to stabilize the trajectory. Time series of length $T = 1000$ were then simulated with step size $\Delta t = 0.1$.

To introduce non-stationarity in the dynamics, the NMDA conductance parameter g_{NMDA} was varied across simulations. Three values were used to generate qualitatively distinct regimes: $g_{NMDA} = 9.3$, 10.3, and 11.3. For each value, one time series of states $[V(t), n(t), h(t)]^\top$ was obtained.

The pePLRNN model for the NMDA Burster experiment was trained using the following parameters: number of training epochs = 100001, initial teacher forcing parameter `TF_alpha` = 0.5, final teacher forcing value `TF_alpha2` = 10^{-20}, number of hidden dimensions $M = 64$, L2 regularization coefficient $\lambda_1 = 10^{-5}$, temporal smoothness regularization $\lambda_2 = 64$, batch size = 400, sequence length = 80, and number of independent runs = 20.

Generation of Benchmark Data from the Logistic Map

Benchmark time series data of the logistic map was generated under different dynamical regimes. The logistic map is defined by:

$$x_{t+1} = rx_t(1 - x_t), \tag{30}$$

where $x_t \in [0, 1]$ and $r \in [0, 4]$ is the bifurcation control parameter.

Time series of length $T = 5000$ were generated for four distinct parameter values $r \in \{2.8, 3.2, 3.5, 3.9\}$, corresponding to qualitatively different dynamical regimes (one fixed point, 2-cycle, 4-cycle and chaos). A fixed initial condition $x_0 = 0.5$ was used across all simulations.

The pePLRNN model for the Logistic Map experiment was trained using the following parameters: number of training epochs = 20001, initial teacher forcing parameter `TF_alpha` = 0.5, final teacher forcing value `TF_alpha2` = 10^{-20}, number of hidden dimensions $M = 1$, L2 regularization coefficient $\lambda_1 = 10^{-5}$, temporal smoothness regularization values $\lambda_2 = 64$, batch size = 256, sequence length = 2, and number of independent runs = 10.

In addition, a bifurcation dataset was generated to test the models ability to capture transitions in qualitative dynamics across a range of r values. For this, 50 values of r were linearly spaced between $r_{\min} = 2.5$ and $r_{\max} = 4.0$. For each r, the system was iterated for 1000 transient steps followed by 100 iterations used for model training. The resulting state vectors $\{x_t\}$ were stored for each r creating the bifurcation dataset.

The pePLRNN model for the bifurcation dataset of the logistic map was trained using the following parameters: number of training epochs = 60001, initial teacher forcing parameter `TF_alpha` = 0.5, final teacher forcing value `TF_alpha2` = 10^{-20}, number of hidden dimensions $M = 8$, L2 regularization coefficient $\lambda_1 = 10^{-5}$, temporal smoothness regularization $\lambda_2 = 0.1$, batch size = 2048, sequence length = 2, and number of independent runs = 10.

Task-Trained RNN Reconstruction

Task-trained RNNs for the reconstruction experiment of Section 3.1.1 and Section 3.1.1 were trained with $d_{hidden} = 3$, all other specification apply as in Section 2.4.

The best-performing pair of task-trained RNN and its corresponding pePLRNN reconstruction were selected based on a combined evaluation of trajectory reconstruction correlation and behavioral accuracy.

Specifically, for each model pair, I computed the mean correlation between true and reconstructed hidden state trajectories on both the training and test sets, and the mean behavioral prediction error of the task-trained RNN as well as the behavioral prediction error of the reconstruction model.

The final score combined these measures by summing the two correlation values and subtracting the four behavioral errors. The model pair with the highest resulting score was selected.

The pePLRNN model for reconstructing the task-trained RNN dynamics in the first validation experiment was fitted using the following parameters: number of training epochs = 50001, initial teacher forcing parameter `TF_alpha` = 0.5, final teacher forcing value `TF_alpha2` = 10^{-20}, number of hidden dimensions $M = 10$, L2 regularization coefficient $\lambda_1 = 10^{-5}$, temporal smoothness regularization $\lambda_2 = 64$, batch size = 200, sequence length = 20, and number of independent runs = 10.

Task-Trained RNN Reconstruction With and Without Memory

Task-trained RNNs for the reconstruction experiment of Section3.1.2 and section were trained with $d_{hidden} = 8$, with all other specification applying as in Section2.4.

The pePLRNN model for reconstructing the task-trained RNN dynamics comparing with and without memory requirement was fitted using the following parameters: number of training epochs = 50001, initial teacher forcing parameter `TF_alpha` = 0.5, final teacher forcing value `TF_alpha2` = 10^{-20}, number of hidden dimensions $M = 10$, L2 regularization coefficient $\lambda_1 = 10^{-5}$, temporal smoothness regularization $\lambda_2 = 64$, batch size = 200, sequence length = 20, and number of independent runs = 10.

Cross-Condition Reconstruction Experiments

For the cross-condition experiment in Section3.1.3. The hidden state dynamics from taken from Section3.1.2 and remained unchanged. Inputs from the memory task variant were converted to continues inputs by adding 1 to the cue input channel active in the respective trial. Continues inputs were converted to impluse inputs by subtracting 1 from the active cue input channel for the last second.

The pePLRNN model for the reconstruction of task-trained RNNs in the swapped input experiment was trained with the following parameters: number of training epochs = 100001, initial teacher forcing parameter `TF_alpha` = 0.5, final teacher forcing value `TF_alpha2` = 10^{-20}, number of hidden dimensions $M = 10$, L2 regularization coefficient $\lambda_1 = 10^{-5}$, temporal smoothness regularization $\lambda_2 = 64$, batch size = 200, sequence length = 20, and number of independent runs = 10.

Reconstructions from Recorded Data

For all experimental sessions, spike time were convolved using the procedure described in Section2.1. External inputs (see Sect. 2.2) were constructed for every trial separately as one-hot encoding signaling the cue light presentation (left cue and right cue as separate input channels), the lever presentation (as one additional input channel) and reward presentation (as another input channel).

All reconstructions from recorded neural data were performed using the pePLRNN model configured with the following parameters: number of training epochs = 100000, initial teacher forcing parameter `TF_alpha` = 0.5, final teacher forcing value `TF_alpha2` = 0.001, number of hidden dimensions $M = 768$, L2 regularization coefficient $\lambda_1 = 10^{-5}$, temporal smoothness regularization $\lambda_2 = 128$, batch size = 256, sequence length = 400, and number of independent runs = 10.

To generate simulations for evaluating model-generated trajectories on unseen data, animal datasets were reduced to ten trials preceding the rule change point. Half of the stable behavioral trials (10 trials) were held out as test data. Models were trained on the reduced dataset with the same configuration as described in Section 5. To evaluate performance, the trained models were used to generate trajectories for the first five held-out trials. Trajectories were generated by providing the trial-specific initial condition and external input sequence.

True and generated trajectories were compared using unit- and trial-wise Pearson correlation. These values were averaged across units and trials to obtain dataset-specific correlation estimates. A comparison distribution was generated by computing trial-wise cross-correlations (each unit signal being correlated with corresponding unit signal from the other trial) for all pairs among the five held-out trials. Distributions were compared using a Wilcoxon rank-sum test.

Generation of Transients

To generate transient trajectories for different rule conditions in Section 3.2.2, the following procedure was used. For each behaviorally stable period (see Section 2.1), neural states from one second before cue onset were extracted from the respective trials. The sample mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ were estimated from these states. From the resulting Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, 500 initial states $\mathbf{z}_0^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ were sampled for each rule condition.

These initial conditions were propagated for 500 time steps without input using the pePLRNN, with $\mathbf{W}_1$, $\mathbf{W}_2$ set to the average parameter values of the respective behaviorally stable period. The resulting state at step 500, $\mathbf{z}_{500}^{(i)}$, served as the initial condition for generating full trial transients. From each generated initial state, cue-specific trajectories were then generated by applying the cue input vector $\mathbf{s}_{\text{cue}}$ for 60 time steps, followed by a choice input vector $\mathbf{s}_{\text{choice}}$. The duration of the choice input was set to the mean reaction time observed in the respective rule condition.

Influence of Rule-Specific Initial Condition and Parameters

To assess the relative influence of initial conditions and parameters on transient trajectories in Section 3.2.3, the same procedure described in Section **??** was used. However, prior to input-driven trial generation, the initial conditions $\mathbf{z}_{500}^{(i)}$ were swapped between rule conditions. Swapped initial state was then evolved with the parameter set $\mathbf{W}_1, \mathbf{W}_2$) of the respective opposite rule for both cue conditions.

Generation of Cue-Driven Long-Term Simulations

To generate cue-specific limit sets, the same initial condition sampling procedure was used as described in Section **??**. For each behaviorally stable period (see Section 2.1), the sample mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ were estimated from neural states one second prior to cue onset. From the resulting Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, 500 initial latent states $\mathbf{z}_0^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ were sampled per rule condition.

Each initial state was first propagated for 500 time steps without input using the pePLRNN with parameter matrices $\mathbf{W}_1$, $\mathbf{W}_2$ set to the average values of the respective behavioral regime, resulting in the initial state $\mathbf{z}_{500}^{(i)}$.

From these initial states, cue-specific long-term trajectories were generated by applying the corresponding cue input vector $\mathbf{s}_{\text{cue}}$ continuously for 5000 time steps.

Generation of Long-Term Simulations to Test for Multistability

To generate the limit sets for Section 3.2.3, the final states of previously generated cue transients and cue long-term generatedtrajectories (see Sections **??** and **??**) were

taken as initial conditions. Each of these states was then propagated for 20000 time steps without any external input.

Trial-by-Trial Trajectory Generation

To generate trajectories on a trial-by-trial basis for Section3.2.4, the same procedure described in Sections **??** and **??** was followed, but instead of each behaviorally stable period each trial was considered. For each trial, the corresponding parameters $\mathbf{W}_1^{(k)}$, $\mathbf{W}_2^{(k)}$ assigned to that trial were used to simulate the dynamics. Initial conditions were computed individually for each trial like in Sections **??**.

Time Series for Change Point Analysis

For change point analyses in Section3.2.4 of the trial-specific parameters, locations of attracting regions, and recorded neural states, the time series, fitted by PARCS (see Sect. 2.5), was considered from the onset of the behaviorally stable period of the first rule condition to the end of the stable period of the second rule for the respective experimental session. For the determination of the behavioral change point as specified in Section2.5. The trial-specific behavioral responses after the rule change until the end of the respective experimental session were considered.

Connectivity estimation

For analysis of connectivity similarity, connectivity was estimate for every trial by simulating 500 trajectories (described in section2.5). These connectivity estimates were then thresholded to extract the 5% weights (all other were set to zero). Jaccard similarty was used to calculate the similarity of effective similarity across trials. To compute clusters, upper triangles of similarity matrices were extracted and separated by the CP. Shuffle comparison was done by randomly sampling the CP.

Rule Learning in Each Session

Correct implementation of rules is tested with chi-squared table test. VR is learned if chi-squared test is significant such that there is a significant relation between cue and choice. The SR is learned if there is no significant relation between cue and choice.

Table 1: Behavioral Table. Asterisks (*) indicate incorrect behavior.

Dataset	**Rule (1,2)**	**Cue rule learned**	**SR learned**
01	excluded	-	-
02	left, cue	0.0070*	0.4561*
03*	cue, left	0.0935	0.2104*
04*	left, cue	0.0049*	0.0147
05	cue, left	0.0001*	1.0000*
06	cue, left	0.0225*	0.3563*
07*	left, cue	0.1441	1.0000*
08	cue, left	0.0389*	0.1160*
09	left, cue	0.0389*	1.0000*
10	cue, left	0.1047	0.1376*
11	cue, right	0.0014*	1.0000*
12	right, cue	0.0049*	0.5490*
13	cue, right	0.0078*	0.2846*
14	cue, right	0.0014*	0.2104*
15	right, cue	0.0049*	0.5030*
16	cue, right	0.0022	1.0000*
17	excluded	-	-
18	cue, right	0.0014*	0.2104*
19	cue, left	0.0013*	0.2104*
20	left, cue	0.0049*	0.2846*
21	cue, left	0.0016*	0.2104*
22	left, cue	0.0045*	0.2104*
23	cue, left	0.0017*	0.4533*
24*	left, cue	0.7952	0.2846*

Proof of Bounded Orbits in a Single Sub-Network

The proof presented below is adapted and expanded from the framework described in [104]. If $\rho(\mathbf{A}) = \|\mathbf{A}\| < 1$, then every orbit of the clipped shallow PLRNN sub-network is bounded.

Proof. We consider the dynamics of a single sub-network governed by

$$\mathbf{z}_t = \mathbf{A}\,\mathbf{z}_{t-1} + \mathbf{W}_1\left[\phi\big(\mathbf{W}_2\,\mathbf{z}_{t-1} + \mathbf{h}_2\big) - \phi\big(\mathbf{W}_2\,\mathbf{z}_{t-1}\big)\right] + \mathbf{h}_1, \tag{31}$$

where $\mathbf{z}_t \in \mathbb{R}^M$ is the state vector at time t, $\mathbf{A} \in \mathbb{R}^{M\times M}$ is a diagonal matrix, and $\mathbf{W}_1 \in \mathbb{R}^{M\times L}$, $\mathbf{W}_2 \in \mathbb{R}^{L\times M}$ are connectivity matrices. The bias vectors are $\mathbf{h}_1 \in \mathbb{R}^M$ and $\mathbf{h}_2 \in \mathbb{R}^L$, and the nonlinearity $\phi(\cdot)$ is the ReLU function.

For notational convenience, we define

$$\psi(\mathbf{z}_{t-1}) \coloneqq \phi\big(\mathbf{W}_2\,\mathbf{z}_{t-1} + \mathbf{h}_2\big) - \phi\big(\mathbf{W}_2\,\mathbf{z}_{t-1}\big).$$

Thus, equation (31) can be rewritten as

$$\mathbf{z}_t = \mathbf{A}\,\mathbf{z}_{t-1} + \mathbf{W}_1\,\psi(\mathbf{z}_{t-1}) + \mathbf{h}_1. \tag{32}$$

Step 1: Bounding the Nonlinear Term. For each component $l \in \{1,\dots,L\}$, by definition,

$$\psi_l(\mathbf{z}_{t-1}) = \max\Big\{0, \sum_{j=1}^{M} w_{lj}^{(2)}\, z_{j,t-1} + h_2^{(l)}\Big\} - \max\Big\{0, \sum_{j=1}^{M} w_{lj}^{(2)}\, z_{j,t-1}\Big\}.$$

Since the ReLU function is non-decreasing, it follows directly that

$$\psi_l(\mathbf{z}_{t-1}) \le h_2^{(l)}, \quad \text{for all } l = 1,\dots,L. \tag{33}$$

Taking the Euclidean norm, we deduce

$$\|\psi(\mathbf{z}_{t-1})\| = \sqrt{\sum_{l=1}^{L} \psi_l(\mathbf{z}_{t-1})^2} \le \sqrt{L}\, h_{\max} \equiv \bar{h}_{\max}, \tag{34}$$

where

$$h_{\max} = \max_{1\le l\le L} h_2^{(l)}.$$

Step 2: Recursive Estimation of the Orbit. Let $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T, \dots\}$ be an orbit of the dynamics (32). By recursive substitution we obtain

$$\begin{aligned}
\mathbf{z}_2 &= \mathbf{A}\,\mathbf{z}_1 + \mathbf{W}_1\,\psi(\mathbf{z}_1) + \mathbf{h}_1,\\
\mathbf{z}_3 &= \mathbf{A}^2\,\mathbf{z}_1 + \mathbf{A}\,\mathbf{W}_1\,\psi(\mathbf{z}_1) + \mathbf{W}_1\,\psi(\mathbf{z}_2) + (\mathbf{A}+\mathbf{I})\,\mathbf{h}_1,\\
&\;\;\vdots\\
\mathbf{z}_T &= \mathbf{A}^{T-1}\,\mathbf{z}_1 + \sum_{j=0}^{T-2} \mathbf{A}^j\,\mathbf{W}_1\,\psi\big(\mathbf{z}_{T-1-j}\big) + \sum_{j=0}^{T-2} \mathbf{A}^j\,\mathbf{h}_1.
\end{aligned} \tag{35}$$

Taking norms on both sides and applying the triangle inequality gives

$$\|\mathbf{z}_T\| \leq \|\mathbf{A}\|^{T-1}\|\mathbf{z}_1\| + \|\mathbf{W}_1\| \sum_{j=0}^{T-2} \|\mathbf{A}\|^j \, \|\psi(\mathbf{z}_{T-1-j})\| + \|\mathbf{h}_1\| \sum_{j=0}^{T-2} \|\mathbf{A}\|^j. \tag{36}$$

Using the bound from (34), we have

$$\|\mathbf{z}_T\| \leq \|\mathbf{A}\|^{T-1}\|\mathbf{z}_1\| + \bar{h}_{\max} \|\mathbf{W}_1\| \sum_{j=0}^{T-2} \|\mathbf{A}\|^j + \|\mathbf{h}_1\| \sum_{j=0}^{T-2} \|\mathbf{A}\|^j. \tag{37}$$

Step 3: Convergence of the Series. Since $\|\mathbf{A}\| < 1$, the term $\|\mathbf{A}\|^{T-1}$ decays to zero as $T \to \infty$, and the geometric series $\sum_{j=0}^{\infty} \|\mathbf{A}\|^j$ converges to $\frac{1}{1-\|\mathbf{A}\|}$. Therefore, taking the limit as $T \to \infty$ in (37) yields

$$\lim_{T\to\infty} \|\mathbf{z}_T\| \leq \bar{h}_{\max} \|\mathbf{W}_1\| \frac{1}{1-\|\mathbf{A}\|} + \|\mathbf{h}_1\| \frac{1}{1-\|\mathbf{A}\|} < \infty. \tag{38}$$

This result establishes that every orbit of the sub-network described by (32) remains bounded under the condition $\|\mathbf{A}\| < 1$. □

Proof for Whittaker-Handerson Smoother

We have a sequence of parameter matrices $\{\mathbf{W}_1^{(k)}, \mathbf{W}_2^{(k)}\}_{k=1}^K$. Define

$$\mathbf{x}^{(k)} \;=\; \begin{pmatrix} \mathrm{vec}\big(\mathbf{W}_1^{(k)}\big) \\ \mathrm{vec}\big(\mathbf{W}_2^{(k)}\big) \end{pmatrix} \in \mathbb{R}^n,$$

so that the given penalty

$$\frac{\lambda_2}{2K} \sum_{k=2}^{K} \Big\|\mathbf{W}_1^{(k)} - \mathbf{W}_1^{(k-1)}\Big\|_F^2 \;+\; \Big\|\mathbf{W}_2^{(k)} - \mathbf{W}_2^{(k-1)}\Big\|_F^2$$

can be rewritten as

$$\frac{\lambda_2}{2K} \sum_{k=2}^{K} \big\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\big\|_2^2 \;=\; \frac{\lambda_2}{2K} \, \|D\,X\|_F^2,$$

where

$$X = \begin{pmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(K)} \end{pmatrix}, \quad D = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix}.$$

This is *exactly* the same form as the discrete Whittaker–Henderson penalty $\sum_t \|x_t - x_{t-1}\|^2$. Hence the entire objective

$$\min_X \; \|Y - X\|_F^2 \;+\; \frac{\lambda_2}{2K} \, \|D\,X\|_F^2$$

is a matrix-valued analogue of the Whittaker smoother $\min_x \|y - x\|_2^2 + \lambda \|Dx\|_2^2$.

2. Proof of Zero-Phase Property

We now show that the resulting smoother operator is symmetric and hence zero-phase.

(i) Define the *smoother matrix* on each row-vector of X by

$$S \;=\; (I + \alpha\, D^T D)^{-1}, \qquad \alpha = \frac{\lambda_2}{2K}\,.$$

Then the fitted sequence is $\widehat{X} = S\,Y$.

(ii) Note that

$$(D^T D)^T \;=\; D^T\,(D^T)^T \;=\; D^T D,$$

so $D^T D$ is symmetric.

(iii) The identity matrix I is symmetric and for any scalar α, so is $I + \alpha\, D^T D$. Call this sum A.

(iv) If A is symmetric and invertible, then

$$A^{-1} \;=\; (A^T)^{-1} \;=\; (A^{-1})^T,$$

so A^{-1} is also symmetric.

(v) Hence

$$S \;=\; (I + \alpha\, D^T D)^{-1}$$

is symmetric. Any linear operator with a symmetric matrix representation has *zero-phase*: it treats past and future data identically and introduces no net time-shift of CPs.

References

[1] Aamir Abbasi, Rohit Rangwani, Daniel W. Bowen, Andrew W. Fealy, Nathan P. Danielsen, and Tanuj Gulati. "Cortico-Cerebellar Coordination Facilitates Neuroprosthetic Control". In: *Science Advances* 10.15 (2024), eadm8246. DOI: 10.1126/sciadv.adm8246.

[2] Kyle Aitken, Marina Garrett, Shawn Olsen, and Stefan Mihalas. "The geometry of representational drift in natural and artificial neural networks". In: *PLOS Computational Biology* 18.11 (2022), e1010716. DOI: 10.1371/journal.pcbi.1010716.

[3] Lars Albantakis and Gustavo Deco. "The encoding of alternatives in multiple-choice decision making". In: *Proceedings of the National Academy of Sciences of the United States of America* 106.25 (2009), pp. 10308–10313. DOI: 10.1073/pnas.0900814106.

[4] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: An Introduction to Dynamical Systems*. New York: Springer, 1996. ISBN: 978-0-387-94677-5.

[5] Daniel J. Amit. *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge, UK: Cambridge University Press, 1989. ISBN: 978-0-521-36100-2.

[6] Vasso Anagnostopoulou, Christian Pötzsche, and Martin Rasmussen. *Nonautonomous Bifurcation Theory: Concepts and Tools*. Vol. 10. Frontiers in Applied Dynamical Systems: Reviews and Tutorials. Cham: Springer, 2023. ISBN: 978-3-031-29841-7. DOI: 10.1007/978-3-031-29842-4.

[7] Paul G. Anastasiades and Adam G. Carter. "Circuit organization of the rodent medial prefrontal cortex". In: *Trends in Neurosciences* 44.7 (2021), pp. 550–563. DOI: 10.1016/j.tins.2021.03.006.

[8] Makoto C. Aoi, Valerio Mante, and Jonathan W. Pillow. "Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making". In: *Nature Neuroscience* 23 (2020), pp. 1410–1420. DOI: 10.1038/s41593-020-0696-5.

[9] Lisa Aziz-Zadeh, Jason T. Kaplan, and Marco Iacoboni. ""Aha!": The Neural Correlates of Verbal Insight Solutions". In: *Human Brain Mapping* 30.3 (2009), pp. 908–916. DOI: 10.1002/hbm.20554.

[10] Florian Bähner et al. "Species-conserved mechanisms of cognitive flexibility in complex environments". In: *bioRxiv* (2022). DOI: 10.1101/2022.11.14.516439.

[11] Omri Barak. "Recurrent Neural Networks as Versatile Tools of Neuroscience Research". In: *Current Opinion in Neurobiology* 46 (2017), pp. 1–6. DOI: 10.1016/j.conb.2017.06.003.

[12] Manuel Beiran, Nicolas Meirhaeghe, Hansem Sohn, Mehrdad Jazayeri, and Srdjan Ostojic. "Parametric control of flexible timing through low-dimensional neural manifolds". In: *Neuron* 111.5 (2023), pp. 739–753.

[13] Yoshua Bengio. "Learning Deep Architectures for AI". In: *Foundations and Trends in Machine Learning* 2.1 (2009), pp. 1–127. DOI: 10.1561/2200000006.

[14] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning Long-Term Dependencies with Gradient Descent is Difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[15] Lucy K Bicks, Hiroyuki Koike, Schahram Akbarian, and Hirofumi Morishita. "Prefrontal cortex and social cognition in mouse and man". In: *Frontiers in psychology* 6 (2015), p. 1805.

[16] Joanna M. Birrell and Veronica J. Brown. "Medial frontal cortex mediates perceptual attentional set shifting in the rat". In: *Journal of Neuroscience* 20.11 (2000), pp. 4320–4324. DOI: 10.1523/JNEUROSCI.20-11-04320.2000.

[17] Gregory B. Bissonette, A. G. Powell, and A. L. Roesch. "Double dissociation of the effects of medial and orbital prefrontal cortical lesions on attentional and affective shifts in mice". In: *Journal of Neuroscience* 28.44 (2008), pp. 11124–11130.

[18] Amy E. Block, Hooman Dhanji, Sarah F. Thompson-Tardif, and Stan B. Floresco. "Thalamic-prefrontal cortical-ventral striatal circuitry mediates dissociable components of strategy set shifting". In: *Cerebral Cortex* 17 (2007), pp. 1625–1636. DOI: 10.1093/cercor/bhl073.

[19] Bastiaan Bloem, Rogier B. Poorthuis, and Huibert D. Mansvelder. "Cholinergic modulation of the medial prefrontal cortex: the role of nicotinic receptors in attention and regulation of neuronal activity". In: *Frontiers in Neural Circuits* 8 (2014), p. 17. DOI: 10.3389/fncir.2014.00017.

[20] Georg Bohlmann. "Ein Ausgleichungsproblem". In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* (1899), pp. 260–271.

[21] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. "Sliced and Radon Wasserstein Barycenters of Measures". In: *J. Math. Imaging Vis.* 51.1 (2015), pp. 22–45.

[22] Matthew M. Botvinick, Jonathan D. Cohen, and Cameron S. Carter. "Conflict monitoring and anterior cingulate cortex: An update". In: *Trends in Cognitive Sciences* 8.12 (2004), pp. 539–546. DOI: 10.1016/j.tics.2004.10.003.

[23] Manuel Brenner, Christoph Jürgen Hemmer, Zahra Monfared, and Daniel Durstewitz. "Almost-Linear RNNs Yield Highly Interpretable Symbolic Codes in Dynamical Systems Reconstruction". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2024.

[24] Manuel Brenner, Florian Hess, Jonas M Mikhaeil, Leonard F Bereska, Zahra Monfared, Po-Chen Kuo, and Daniel Durstewitz. "Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 2292–2320.

[25] Manuel Brenner, Georgia Koppe, and Daniel Durstewitz. "Multimodal Teacher Forcing for Reconstructing Nonlinear Dynamical Systems". In: *When Machine Learning meets Dynamical Systems: Theory and Applications*. 2023.

[26] Manuel Brenner, Elias Weber, Georgia Koppe, and Daniel Durstewitz. "Learning Interpretable Hierarchical Dynamical Systems Models from Time Series Data". In: *The Thirteenth International Conference on Learning Representations (ICLR)*. 2025.

[27] Manuel Benjamin Brenner. "Learning Interpretable Dynamical Systems Models from Multimodal Empirical Time Series". Supervised by Prof. Dr. Daniel Durstewitz. PhD thesis. Heidelberg University, 2024. DOI: 10.11588/heidok.00035092.

[28] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937. DOI: 10.1073/pnas.1517384113.

[29] Yoram Burak and Ila R. Fiete. "Accurate Path Integration in Continuous Attractor Network Models of Grid Cells". In: *PLoS Computational Biology* 5.2 (2009), e1000291. DOI: 10.1371/journal.pcbi.1000291.

[30] Timothy J. Buschman and Earl K. Miller. "Goal-direction and top-down control". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 369.1655 (2014), p. 20130471. DOI: 10.1098/rstb.2013.0471.

[31] György Buzsáki and Andreas Draguhn. "Neuronal Oscillations in Cortical Networks". In: *Science* 304.5679 (2004), pp. 1926–1929. DOI: 10.1126/science.1099745.

[32] Francesco Carnevale, Victor de Lafuente, Ranulfo Romo, Omri Barak, and Natalia Parga. "Dynamic Control of Response Criterion in Premotor Cortex during Perceptual Detection under Temporal Uncertainty". In: *Neuron* 86.4 (2015), pp. 1067–1077. DOI: 10.1016/j.neuron.2015.03.031.

[33] Andrea Ceni, Peter Ashwin, and Lorenzo Livi. "Interpreting Recurrent Neural Networks Behaviour via Excitable Network Attractors". In: *Cognitive Computation* 12.2 (2020), pp. 388–404. DOI: 10.1007/s12559-019-09634-2.

[34] Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. "Data-driven discovery of coordinates and governing equations". In: *Proceedings of the National Academy of Sciences* 116.45 (2019), pp. 22445–22451. DOI: 10.1073/pnas.1906995116.

[35] Rishidev Chaudhuri and Ila R. Fiete. "Computational Principles of Memory". In: *Nature Neuroscience* 19.3 (2016), pp. 394–403. DOI: 10.1038/nn.4237.

[36] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. "Neural ordinary differential equations". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018, pp. 6571–6583.

[37] Tianping Chen and Hong Chen. "Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems". In: *IEEE Transactions on Neural Networks* 6.4 (1995), pp. 911–917. DOI: 10.1109/72.392253.

[38] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 103–111.

[39] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[40] Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. "Neural Population Dynamics During Reaching". In: *Nature* 487.7405 (2012), pp. 51–56. DOI: 10.1038/nature11129.

[41] John P. Cunningham and Byron M. Yu. "Dimensionality reduction for large-scale neural recordings". In: *Nature Neuroscience* 17.11 (2014), pp. 1500–1509. DOI: 10.1038/nn.3776.

[42] Clayton E. Curtis and Mark D'Esposito. "Persistent activity in the prefrontal cortex during working memory". In: *Trends in Cognitive Sciences* 7.9 (2003), pp. 415–423. DOI: 10.1016/j.tics.2003.08.009.

[43] George Cybenko. "Approximation by Superpositions of a Sigmoidal Function". In: *Mathematics of Control, Signals, and Systems* 2.4 (1989), pp. 303–314. DOI: 10.1007/BF02551274.

[44] Jeffrey W. Dalley, Rudolf N. Cardinal, and Trevor W. Robbins. "Prefrontal executive and cognitive functions in rodents: Neural and neurochemical substrates". In: *Neuroscience & Biobehavioral Reviews* 28.7 (2004), pp. 771–784. DOI: 10.1016/j.neubiorev.2004.09.006.

[45] Geoffroy Delamare, Yosif Zaki, Denise J. Cai, and Claudia Clopath. "Drift of Neural Ensembles Driven by Slow Fluctuations of Intrinsic Excitability". In: *eLife* 12 (2024), RP88053. DOI: 10.7554/eLife.88053.

[46] Alain Destexhe and Michelle Rudolph-Lilith. *Neuronal Noise*. Vol. 8. Springer Series in Computational Neuroscience. New York, NY: Springer, 2012. ISBN: 978-0-387-79019-0. DOI: 10.1007/978-0-387-79020-6.

[47] Kenji Doya. "Complementary roles of basal ganglia and cerebellum in learning and motor control". In: *Current Opinion in Neurobiology* 10.6 (2000), pp. 732–739. DOI: 10.1016/S0959-4388(00)00153-7.

[48] Laura N. Driscoll, Lea Duncker, and Christopher D. Harvey. "Representational drift: Emerging theories for continual learning and experimental future directions". In: *Current Opinion in Neurobiology* 76 (2022), p. 102609. DOI: 10.1016/j.conb.2022.102609.

[49] Laura N. Driscoll, Krishna Shenoy, and David Sussillo. “Flexible Multitask Computation in Recurrent Networks Utilizes Shared Dynamical Motifs”. In: *Nature Neuroscience* 27.7 (2024), pp. 1349–1363. DOI: 10.1038/s41593-024-01668-6.

[50] Lauren N. Driscoll, Nicholas L. Pettit, Matthias Minderer, Selmaan N. Chettih, and Christopher D. Harvey. “Dynamic reorganization of neuronal activity patterns in parietal cortex”. In: *Cell* 170.5 (2017), 986–999.e16. DOI: 10.1016/j.cell.2017.07.021.

[51] Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. “The Role of Population Structure in Computations Through Neural Dynamics”. In: *Nature Neuroscience* 25.6 (2022), pp. 783–794. DOI: 10.1038/s41593-022-01088-4.

[52] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. “Augmented Neural ODEs”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 3134–3144.

[53] Daniel Durstewitz. “A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements”. In: *PLOS Computational Biology* 13.6 (June 2017), pp. 1–33. DOI: 10.1371/journal.pcbi.1005542.

[54] Daniel Durstewitz. *Advanced Data Analysis in Neuroscience: Integrating Statistical and Computational Models*. Bernstein Series in Computational Neuroscience. Cham: Springer, 2017. ISBN: 978-3-319-59974-8. DOI: 10.1007/978-3-319-59976-2.

[55] Daniel Durstewitz. “Implications of Synaptic Biophysics for Recurrent Network Dynamics and Active Memory”. In: *Neural Networks* 22.8 (2009), pp. 1189–1200. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2009.07.016.

[56] Daniel Durstewitz. “Implications of synaptic biophysics for recurrent network dynamics and active memory”. In: *Neural Networks* 22.8 (2009). Cortical Microcircuits, pp. 1189–1200. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2009.07.016.

[57] Daniel Durstewitz and Gustavo Deco. “Computational significance of transient dynamics in cortical networks”. In: *European Journal of Neuroscience* 27.1 (2008), pp. 217–227. DOI: 10.1111/j.1460-9568.2007.05976.x.

[58] Daniel Durstewitz and Thomas Gabriel. “Dynamical Basis of Irregular Spiking in NMDA-Driven Prefrontal Cortex Neurons”. In: *Cerebral Cortex* 17.4 (Apr. 2007), pp. 894–908. DOI: 10.1093/cercor/bhk044.

[59] Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. “Reconstructing Computational Dynamics from Neural Measurements with Recurrent Neural Networks”. In: *bioRxiv* (2022). DOI: 10.1101/2022.10.31.514408.

[60] Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. “Reconstructing Computational Dynamics from Neural Measurements with Recurrent Neural Networks”. In: *Nature Reviews Neuroscience* (Oct. 4, 2023). DOI: https://doi.org/10.1038/s41583-023-00740-7. published.

[61] Daniel Durstewitz, Jeremy K. Seamans, and Terrence J. Sejnowski. "Neurocomputational models of working memory". In: *Nature Neuroscience* 3.Suppl 11 (2000), pp. 1184–1191. DOI: 10.1038/81460.

[62] Daniel Durstewitz, Nicole M. Vittoz, Stan B. Floresco, and Jeremy K. Seamans. "Abrupt Transitions between Prefrontal Neural Ensemble States Accompany Behavioral Transitions during Rule Learning". In: *Neuron* 66.3 (2010), pp. 438–448. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2010.03.029.

[63] Lukas Eisenmann, Zahra Monfared, Niclas Alexander Göring, and Daniel Durstewitz. "Bifurcations and loss jumps in RNN training". In: *NeurIPS 2023*. Nov. 6, 2023. published.

[64] Jeffrey L. Elman. "Finding Structure in Time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. ISSN: 0364-0213. DOI: 10.1016/0364-0213(90)90002-E.

[65] Rainer Engelken, Fred Wolf, and L. F. Abbott. "Lyapunov Spectra of Chaotic Recurrent Neural Networks". In: *Physical Review Research* 5.4 (2023), p. 043044. DOI: 10.1103/PhysRevResearch.5.043044.

[66] G. Bard Ermentrout and David H. Terman. *Mathematical Foundations of Neuroscience*. Vol. 35. Interdisciplinary Applied Mathematics. New York: Springer, 2010. ISBN: 978-0387877075.

[67] David R. Euston, Aaron J. Gruber, and Bruce L. McNaughton. "The role of medial prefrontal cortex in memory and decision making". In: *Neuron* 76.6 (2012), pp. 1057–1070. DOI: 10.1016/j.neuron.2012.12.002.

[68] Daniel E. Feldman. "The Spike-Timing Dependence of Plasticity". In: *Neuron* 75.4 (2012), pp. 556–571. DOI: 10.1016/j.neuron.2012.08.001.

[69] Leslie K. Fellows and Martha J. Farah. "Is anterior cingulate cortex necessary for cognitive control?" In: *Brain* 128.4 (2005), pp. 788–796. DOI: 10.1093/brain/awh415.

[70] Xiaoli Feng, Gregory J. Perceval, Wei Feng, and Chao Feng. "High Cognitive Flexibility Learners Perform Better in Probabilistic Rule Learning". In: *Frontiers in Psychology* 11 (2020). Published March 13, 2020, p. 415. DOI: 10.3389/fpsyg.2020.00415.

[71] Stan B. Floresco. "Prefrontal dopamine and behavioral flexibility: Shifting from an "inverted-U" toward a family of functions". In: *Frontiers in Neuroscience* 7 (2013), p. 62. ISSN: 1662-453X. DOI: 10.3389/fnins.2013.00062.

[72] Stan B. Floresco, John A. Magyar, Doug M. Ghods-Sharifi, Anthony G. Vexelman, and Anthony A. Tse. "Dissociable roles for the nucleus accumbens core and shell in regulating set shifting". In: *Journal of Neuroscience* 26.9 (2006), pp. 2449–2457. DOI: 10.1523/JNEUROSCI.4431-05.2006.

[73] Stan B. Floresco, Oliver Magyar, Shahrzad Ghods-Sharifi, Corey Vexelman, and Melissa T. Tse. "Multiple dopamine receptor subtypes in the medial prefrontal cortex of the rat regulate set-shifting". In: *Neuropsychopharmacology* 31 (2006), pp. 297–309. DOI: 10.1038/sj.npp.1300825.

[74] Stan B. Floresco, Maric T. L. Tse, and Sarvin Ghods-Sharifi. "Dopaminergic and Glutamatergic Regulation of Effort- and Delay-Based Decision Making". In: *Neuropsychopharmacology* 33.8 (July 2008), pp. 1966–1979. ISSN: 1740-634X. DOI: 10.1038/sj.npp.1301565.

[75] Stan B. Floresco and Melissa T. L. Tse. "Dopaminergic regulation of inhibitory and excitatory transmission in the basolateral amygdala–prefrontal cortical pathway". In: *Journal of Neuroscience* 27.8 (2007), pp. 2045–2057. DOI: 10.1523/JNEUROSCI.5191-06.2007.

[76] Walter J. Freeman. "The Physiology of Perception". In: *Scientific American* 264.2 (1991), pp. 78–85. DOI: 10.1038/scientificamerican0291-78.

[77] Karl Friston. "The free-energy principle: A unified brain theory?" In: *Nature Reviews Neuroscience* 11.2 (2010), pp. 127–138. DOI: 10.1038/nrn2787.

[78] Ken-Ichi Funahashi and Yuichi Nakamura. "Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks". In: *Neural Networks* 6.6 (1993), pp. 801–806. DOI: 10.1016/S0893-6080(05)80125-X.

[79] Joaquín M. Fuster. *The Prefrontal Cortex*. 5th ed. London: Academic Press, 2015. ISBN: 9780124078154.

[80] Joaquín M. Fuster. *The Prefrontal Cortex: Anatomy, Physiology, and Neuropsychology of the Frontal Lobe*. 3rd ed. New York: Lippincott-Raven, 1997. ISBN: 9780781712645.

[81] Joaquín M. Fuster. "Unit activity in prefrontal cortex during delayed-response performance: neuronal correlates of transient memory". In: *Journal of Neurophysiology* 36 (1973), pp. 61–78.

[82] Aniruddh R. Galgali, Maneesh Sahani, and Valerio Mante. "Residual Dynamics Resolves Recurrent Contributions to Neural Computation". In: *Nature Neuroscience* 26.2 (2023), pp. 326–338. DOI: 10.1038/s41593-022-01230-2.

[83] Charles R. Gallistel, Stephen Fairhurst, and Peter Balsam. "The Learning Curve: Implications of a Quantitative Analysis". In: *Proceedings of the National Academy of Sciences* 101.36 (2004), pp. 13124–13131. DOI: 10.1073/pnas.0404965101.

[84] Charles R. Gallistel, Stephen Fairhurst, and Peter Balsam. "The learning curve: Implications of a quantitative analysis". In: *Proceedings of the National Academy of Sciences* 101.36 (2004), pp. 13124–13131. DOI: 10.1073/pnas.0404965101.

[85] Yuanjun Gao, Evan W. Archer, Liam Paninski, and John P. Cunningham. "Linear dynamical neural population models through nonlinear embeddings". In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016, pp. 163–171.

[86] Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. "Toroidal topology of population activity in grid cells". In: *Nature* 602.7895 (2022), pp. 123–128. DOI: 10.1038/s41586-021-04268-7.

[87] J. P. Garner, C. M. Thogerson, H. Würbel, J. D. Murray, and J. A. Mench. "Animal neuropsychology: Validation of the Intra-Dimensional Extra-Dimensional set shifting task for mice". In: *Behavioural Brain Research* 173.1 (2006), pp. 53–61. DOI: 10.1016/j.bbr.2006.06.002.

[88] Patrice Gaspar, Bruno Bloch, and Claude Le Moine. "D1 and D2 receptor gene expression in rat frontal cortex: Cellular localization in different classes of efferent neurons". In: *European Journal of Neuroscience* 7 (1995), pp. 1050–1063. DOI: 10.1111/j.1460-9568.1995.tb01103.x.

[89] Michael Ghil, Mickaël D. Chekroun, and Eric Simonnet. "Climate Dynamics and Fluid Mechanics: Natural Variability and Related Uncertainties". In: *Physica D: Nonlinear Phenomena* 237.14–17 (2008), pp. 2111–2126. DOI: 10.1016/j.physd.2008.03.036.

[90] Michael Ghil and Denisse Sciamarella. "Dynamical systems, algebraic topology and the climate sciences". In: *Nonlinear Processes in Geophysics* 30.4 (2023), pp. 399–434.

[91] Joshua I. Gold and Michael N. Shadlen. "The neural basis of decision making". In: *Annual Review of Neuroscience* 30 (2007), pp. 535–574. DOI: 10.1146/annurev.neuro.29.051605.113038.

[92] Anton Golovanev and Alexander Hvatov. "On the Balance Between the Training Time and Interpretability of Neural ODE for Time Series Modelling". In: *Proceedings of the 9th International Conference on Time Series and Forecasting (ITISE)*. University of Granada, 2022, pp. 61–72.

[93] Matthew D. Golub and David Sussillo. "FixedPointFinder: A Tensorflow toolbox for identifying and characterizing fixed points in recurrent neural networks". en. In: *Journal of Open Source Software* 3.31 (Nov. 2018), p. 1003. ISSN: 2475-9066. DOI: 10.21105/joss.01003.

[94] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[95] Stéphanie Granon and Bruno Poucet. "Involvement of the rat prefrontal cortex in cognitive functions: A central role for the prelimbic area". In: *Psychobiology* 28.2 (2000), pp. 229–237.

[96] D. A. Grant and E. A. Berg. "A behavioral analysis of degree of reinforcement and ease of shifting to new responses in a Weigl-type card-sorting problem". In: *Journal of Experimental Psychology* 38.4 (1948). Original study introducing the Wisconsin Card Sorting Test, foundational for research on cognitive flexibility, pp. 404–411. DOI: 10.1037/h0059831.

[97] Ileana L. Hanganu-Opatz et al. "Resolving the prefrontal mechanisms of adaptive cognitive behaviors: A cross-species perspective". In: *Neuron* 111.7 (2023), pp. 1020–1036. DOI: 10.1016/j.neuron.2023.03.017.

[98] Christopher D. Harvey, Philip Coen, and David W. Tank. "Choice-specific sequences in parietal cortex during a virtual-navigation decision task". In: *Nature* 484.7392 (2012), pp. 62–68. DOI: 10.1038/nature10918.

[99] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley, 1949.

[100] Sarah R Heilbronner, Jose Rodriguez-Romaguera, Gregory J Quirk, Henk J Groenewegen, and Suzanne N Haber. "Circuit-based corticostriatal homologies between rat and primate". In: *Biological psychiatry* 80.7 (2016), pp. 509–521.

[101] Christoph Jürgen Hemmer, Manuel Brenner, Florian Hess, and Daniel Durstewitz. "Optimal Recurrent Network Topologies for Dynamical Systems Reconstruction". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. PMLR, 21–27 Jul 2024, pp. 18174–18204.

[102] Christoph Jürgen Hemmer, Manuel Brenner, Florian Hess, and Daniel Durstewitz. "Optimal Recurrent Network Topologies for Dynamical Systems Reconstruction". In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, 21–27 Jul 2024, pp. 18174–18204.

[103] R. Henderson. "A new method of graduation". In: *Transactions of the Actuarial Society of America* 25 (1924), pp. 29–40.

[104] Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. "Generalized Teacher Forcing for Learning Chaotic Dynamics". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 13017–13049.

[105] Guilherme Shigueto Vilar Higa, Felipe José Costa Viana, José Francis-Oliveira, Emily Cruvinel, Thainá Soares Franchin, Tania Marcourakis, Henning Ulrich, and Roberto De Pasquale. "Serotonergic Neuromodulation of Synaptic Plasticity". In: *Neuropharmacology* (2024). DOI: 10.1016/j.neuropharm.2024.109567.

[106] Sepp Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen". Diplomarbeit. Munich, Germany: Technische Universität München, 1991.

[107] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[108] Alan L Hodgkin and Andrew F Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of physiology* 117.4 (1952), p. 500.

[109] John J Hopfield. "Neural networks and physical systems with emergent collective computational abilities." In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.

[110] John J. Hopfield. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities". In: *Proceedings of the National Academy of Sciences of the United States of America* 79.8 (1982), pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554.

[111] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks are Universal Approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8.

[112] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors". In: *Neural Computation* 25.2 (2013), pp. 328–373. DOI: 10.1162/NECO_a_00393.

[113] Vladimir Itskov, David Hansel, and Misha Tsodyks. "Short-Term Facilitation May Stabilize Parametric Working Memory Trace". In: *Frontiers in Computational Neuroscience* 5 (2011), p. 40. DOI: 10.3389/fncom.2011.00040.

[114] Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007.

[115] Herbert Jaeger and Harald Haas. "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication". In: *Science* 304.5667 (2004), pp. 78–80. DOI: 10.1126/science.1091277.

[116] Max E. Joffe, Chiaki I. Santiago, Julie L. Engers, Craig W. Lindsley, and P. Jeffrey Conn. "Metabotropic glutamate receptor subtype 3 gates acute stress-induced dysregulation of amygdalo-cortical function". In: *Molecular Psychiatry* 24 (2019), pp. 916–927. DOI: 10.1038/s41380-017-0015-z.

[117] Max E. Joffe et al. "Mechanisms underlying prelimbic prefrontal cortex mGlu3/mGlu5-dependent plasticity and reversal learning deficits following acute stress". In: *Neuropharmacology* 144 (2019), pp. 19–28. DOI: 10.1016/j.neuropharm.2018.10.013.

[118] Caroline M. Johnson, Hilary Peckler, Li H. Tai, et al. "Rule learning enhances structural plasticity of long-range axons in frontal cortex". In: *Nature Communications* 7 (2016), p. 10785. DOI: 10.1038/ncomms10785.

[119] Sheena A. Josselyn and Susumu Tonegawa. "Memory Engrams: Recalling the Past and Imagining the Future". In: *Science* 367.6473 (2020), eaaw4325. DOI: 10.1126/science.aaw4325.

[120] Holger Kantz and Thomas Schreiber. *Nonlinear Time Series Analysis*. 2nd ed. Vol. 7. Cambridge Nonlinear Science Series. Cambridge: Cambridge University Press, 2004. ISBN: 978-0521529020.

[121] Nikolaos Karalis and Anton Sirota. "Breathing coordinates cortico-hippocampal dynamics in mice during offline states". In: *Nature Communications* 13.1 (2022), p. 467. DOI: 10.1038/s41467-022-28021-2.

[122] Mattias P. Karlsson, Dougal G. R. Tervo, and Alla Y. Karpova. "Network Resets in Medial Prefrontal Cortex Mark the Onset of Behavioral Uncertainty". In: *Science* 338.6103 (2012), pp. 135–139. DOI: 10.1126/science.1226518.

[123] Steven W. Kennerley, Mark E. Walton, Timothy E. J. Behrens, Mark J. Buckley, and Matthew F. S. Rushworth. "Optimal decision making and the anterior cingulate cortex". In: *Nature Neuroscience* 9.7 (2006), pp. 940–947. DOI: 10.1038/nn1724.

[124] Masahiro Kimura and Ryohei Nakano. "Learning Dynamical Systems by Recurrent Neural Networks from Orbits". In: *Neural Networks* 11.9 (1998), pp. 1589–1599. DOI: 10.1016/S0893-6080(98)00098-7.

[125] Peter E. Kloeden and Martin Rasmussen. *Nonautonomous Dynamical Systems*. Vol. 176. Mathematical Surveys and Monographs. American Mathematical Society, 2011. ISBN: 978-0-8218-6871-3. DOI: 10.1090/surv/176.

[126] Nancy Kopell and G. Bard Ermentrout. "Coupled oscillators and the design of central pattern generators". In: *Mathematical Biosciences* 90 (1988), pp. 87–109. DOI: 10.1016/0025-5564(88)90038-2.

[127] Georgia Koppe, Hazem Toutounji, Peter Kirsch, Stefanie Lis, and Daniel Durstewitz. "Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI". In: *PLOS Computational Biology* 15.8 (Aug. 2019), pp. 1–35. DOI: 10.1371/journal.pcbi.1007263.

[128] Daniel Kramer, Philine L Bommer, Carlo Tombolini, Georgia Koppe, and Daniel Durstewitz. "Reconstructing Nonlinear Dynamical Systems from Multi-Modal Time Series". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 11613–11633.

[129] P. Landry, C. J. Wilson, and S. T. Kitai. "Morphological and Electrophysiological Characteristics of Pyramidal Tract Neurons in the Rat". In: *Experimental Brain Research* 57 (1984), pp. 177–190. DOI: 10.1007/BF00231144.

[130] Kenneth W. Latimer, Jacob L. Yates, Miriam L. R. Meister, Alexander C. Huk, and Jonathan W. Pillow. "Single-trial spike trains in parietal cortex reveal discrete steps during decision-making". In: *Science* 349.6244 (2015), pp. 184–187. DOI: 10.1126/science.aaa4056.

[131] Xiaoxin Liao and Jun Wang. "Global Dissipativity of Continuous-Time Recurrent Neural Networks with Time Delay". In: *Physical Review E* 68.1 (2003), p. 016118. DOI: 10.1103/PhysRevE.68.016118.

[132] M.J. Lighthill and G.B. Whitham. "On kinematic waves. II. A theory of traffic flow on long crowded roads". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 229.1178 (1955), pp. 317–345. DOI: 10.1098/rspa.1955.0089.

[133] Chia-Ying Lin and Chih-Lin Huang. "Considerations for using the Wisconsin Card Sorting Test to assess cognitive flexibility". In: *Journal of Neuroscience Methods* 352 (2021), p. 109089. DOI: 10.1016/j.jneumeth.2021.109089.

[134] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. "On the Variance of the Adaptive Learning Rate and Beyond". en. In: Apr. 2020.

[135] José G. Llavona. *Approximation of Continuously Differentiable Functions*. Vol. 130. North-Holland Mathematics Studies. Amsterdam: North-Holland, 1986. ISBN: 978-0-444-55685-5.

[136] Edward N. Lorenz. "Deterministic Nonperiodic Flow". In: *Journal of the Atmospheric Sciences* 20.2 (1963), pp. 130–141. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[137] Wolfgang Maass, Thomas Natschläger, and Henry Markram. "Computational Models for Generic Cortical Microcircuits". In: *Computational Neuroscience: A Comprehensive Approach*. Ed. by Jianfeng Feng. Chapman & Hall/CRC, 2004, pp. 575–605. ISBN: 978-1584883771.

[138] Wolfgang Maass, Thomas Natschläger, and Henry Markram. "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations". In: *Neural Computation* 14.11 (2002), pp. 2531–2560. DOI: 10.1162/089976602760407955.

[139] Fardad A. Mansouri, Etienne Koechlin, Marcello G. P. Rosa, and Mark J. Buckley. "Managing competing goals–a key role for the frontopolar cortex". In: *Nature Reviews Neuroscience* 18.11 (2017), pp. 645–657. DOI: 10.1038/nrn.2017.111.

[140] Farshad A. Mansouri, Mark J. Buckley, and Keiji Tanaka. "Mnemonic Function of the Dorsolateral Prefrontal Cortex in Conflict-Induced Behavioral Adjustment". In: *Science* 318.5852 (2007), pp. 987–990. DOI: 10.1126/science.1146384.

[141] Farshad A. Mansouri, Kenji Matsumoto, and Keiji Tanaka. "Prefrontal Cell Activities Related to Monkeys' Success and Failure in Adapting to Rule Changes in a Wisconsin Card Sorting Test Analog". In: *The Journal of Neuroscience* 26.10 (2006), pp. 2745–2756. DOI: 10.1523/JNEUROSCI.5238-05.2006.

[142] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. "Context-dependent computation by recurrent dynamics in prefrontal cortex". In: *Nature* 503.7474 (Nov. 2013), pp. 78–84. ISSN: 1476-4687. DOI: 10.1038/nature12742.

[143] Eve Marder. "Neuromodulation of Neuronal Circuits: Back to the Future". In: *Neuron* 76.1 (2012), pp. 1–11. DOI: 10.1016/j.neuron.2012.09.010.

[144] Eve Marder and Dirk Bucher. "Central pattern generators and the control of rhythmic movements". In: *Current Biology* 11.23 (2001), R986–R996. DOI: 10.1016/S0960-9822(01)00581-4.

[145] Stephen J. Martin, Paul D. Grimwood, and Richard G. M. Morris. "Synaptic plasticity and memory: an evaluation of the hypothesis". In: *Annual Review of Neuroscience* 23 (2000), pp. 649–711. DOI: 10.1146/annurev.neuro.23.1.649.

[146] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133. DOI: 10.1007/BF02478259.

[147] Bruce L. McNaughton, Francesco P. Battaglia, Ole Jensen, Edvard I. Moser, and May-Britt Moser. "Path integration and the neural basis of the 'cognitive map'". In: *Nature Reviews Neuroscience* 7.8 (2006), pp. 663–678. DOI: 10.1038/nrn1932.

[148] M.-M. Mesulam. "From sensation to cognition". In: *Brain* 121.6 (1998), pp. 1013–1052. DOI: 10.1093/brain/121.6.1013.

[149] Claus Metzner and Patrick Krauss. *Dynamical Phases and Resonance Phenomena in Information-Processing Recurrent Neural Networks*. 2021.

[150] Claus Metzner and Patrick Krauss. "Dynamics and Information Import in Recurrent Neural Networks". In: *Frontiers in Computational Neuroscience* 16 (2022). ISSN: 1662-5188. DOI: 10.3389/fncom.2022.876315.

[151] Jamilja A. J. van der Meulen, Ruud N. J. M. A. Joosten, Jan P. C. de Bruin, and Matthijs G. P. Feenstra. "Dopamine and Noradrenaline Efflux in the Medial Prefrontal Cortex During Serial Reversals and Extinction of Instrumental Goal-Directed Behavior". In: *Cerebral Cortex* 17.6 (2007), pp. 1444–1453. DOI: 10.1093/cercor/bhl057.

[152] Jonas Mikhaeil, Zahra Monfared, and Daniel Durstewitz. "On the Difficulty of Learning Chaotic Dynamics with RNNs". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 1–12.

[153] Earl K. Miller and Jonathan D. Cohen. "An integrative theory of prefrontal cortex function". In: *Annual Review of Neuroscience* 24.1 (2001), pp. 167–202.

[154] Earl K. Miller, Cheryl A. Erickson, and Robert Desimone. "Neural mechanisms of visual working memory in prefrontal cortex of the macaque". In: *Journal of Neuroscience* 16.16 (1996), pp. 5154–5167. ISSN: 0270-6474.

[155] Paul Miller. "Dynamical systems, attractors, and neural circuits". In: *F1000Research* 5 (2016), F1000. DOI: 10.12688/f1000research.7698.1.

[156] R. K. Miller. "Almost Periodic Differential Equations as Dynamical Systems with Applications to the Existence of Almost Periodic Solutions". In: *Journal of Differential Equations* 1.3 (1965), pp. 337–345. DOI: 10.1016/0022-0396(65)90012-4.

[157] Brenda Milner. "Effects of different brain lesions on card sorting: The role of the frontal lobes". In: *Archives of Neurology* 9.1 (1963), pp. 90–100.

[158] Hannah R. Monday, Thomas J. Younts, and Pablo E. Castillo. "Long-Term Plasticity of Neurotransmitter Release: Emerging Mechanisms and Contributions to Brain Function and Disease". In: *Annual Review of Neuroscience* 41 (2018), pp. 299–322. DOI: 10.1146/annurev-neuro-080317-062155.

[159] Sarah Morceau, Angélique Faugère, Etienne Coutureau, and Mathieu Wolff. "The mediodorsal thalamus supports adaptive responding based on stimulus-outcome associations". In: *Current Research in Neurobiology* 3 (2022), p. 100057. ISSN: 2665-945X. DOI: https://doi.org/10.1016/j.crneur.2022.100057.

[160] Lidija Mrzljak, Clare Bergson, Marie Pappy, Robin Huff, Robert Levenson, and Patricia S. Goldman-Rakic. "Localization of dopamine D4 receptors in GABAergic neurons of the primate brain". In: *Nature* 381 (1996), pp. 245–248. DOI: 10.1038/381245a0.

[161] Edward C. Muly, Krisztina Szigeti, and Patricia S. Goldman-Rakic. "D1 receptor in interneurons of macaque prefrontal cortex: Distribution and subcellular localization". In: *Journal of Neuroscience* 18 (1998), pp. 10553–10565.

[162] Elisabeth A. Murray and Adriana Izquierdo. "Amygdala and orbitofrontal cortex lesions differentially influence choices during object reversal learning". In: *Journal of Neuroscience* 27.32 (2007), pp. 8358–8366. DOI: 10.1523/JNEUROSCI.2279-07.2007.

[163] Elisabeth A. Murray and Peter H. Rudebeck. "Specializations for reward-guided decision-making in the primate ventral prefrontal cortex". In: *Nature Reviews Neuroscience* 19.7 (2018), pp. 404–417. DOI: 10.1038/s41583-018-0013-4.

[164] Y. Nakamura, Y. Nakamura, A. Pelosi, B. Djemai, C. Debacker, D. Herve, et al. "fMRI detects bilateral brain network activation following unilateral chemogenetic activation of direct striatal projection neurons". In: *NeuroImage* 220 (2020), p. 117079. DOI: 10.1016/j.neuroimage.2020.117079.

[165] Nandakumar S. Narayanan and Mark Laubach. "Neuronal correlates of post-error slowing in the rat dorsomedial prefrontal cortex". In: *Journal of Neurophysiology* 100 (2008), pp. 520–525. DOI: 10.1152/jn.00075.2008.

[166] Nandakumar S. Narayanan and Mark Laubach. "Top-down control of motor cortex ensembles by dorsomedial prefrontal cortex". In: *Neuron* 52 (2006), pp. 921–931.

[167] Jérémie Naudé et al. "Dopamine builds and reveals reward-associated latent behavioral attractors". In: *Nature Communications* 15 (2024), p. 9825. DOI: 10.1038/s41467-024-53976-x.

[168] Susan M. Nicola, D. James Surmeier, and Robert C. Malenka. "Dopaminergic modulation of neuronal excitability in the striatum and nucleus accumbens". In: *Annual Review of Neuroscience* 23.1 (2000), pp. 185–215. DOI: 10.1146/annurev.neuro.23.1.185.

[169] Georg Northoff, Niall W. Duncan, and Dave J. Hayes. "The brain and its resting state activity—Experimental and methodological implications". In: *Progress in Neurobiology* 92.4 (2010), pp. 593–600. ISSN: 0301-0082. DOI: 10.1016/j.pneurobio.2010.09.002.

[170] Erik Nyhus and Francisco Barceló. "The Wisconsin Card Sorting Test and the cognitive assessment of prefrontal executive functions: A critical update". In: *Brain and Cognition* 71.3 (2009), pp. 437–451.

[171] Randall C. O'Reilly and Michael J. Frank. "Making Working Memory Work: A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia". In: *Neural Computation* 18.2 (2006), pp. 283–328. DOI: 10.1162/089976606775093909.

[172] Daniel J. O'Shea et al. "Direct neural perturbations reveal a dynamical mechanism for robust computation". In: *bioRxiv* (2022). DOI: 10.1101/2022.12.16.520768.

[173] Leandro A. Oliveira, Taciana R. S. Pollo, Elinéia A. Rosa, Josiane O. Duarte, Carlos H. Xavier, and Carlos C. Crestani. "Both Prelimbic and Infralimbic Noradrenergic Neurotransmissions Modulate Cardiovascular Responses to Restraint Stress in Rats". In: *Frontiers in Physiology* 12 (2021). ISSN: 1664-042X. DOI: 10.3389/fphys.2021.700540.

[174] Ingrid R. Olson, Emily L. Von Der Heide, Jennifer J. Alm, Lindsay J. Vyas, and Sarah C. Tovar-Moll. "Fronto-temporal white matter connectivity predicts reversal learning errors". In: *Cerebral Cortex* 25.12 (2015), pp. 4923–4931. DOI: 10.1093/cercor/bhv134.

[175] Catherine Oualian and Pascale Gisquet-Verrier. "The differential involvement of the prelimbic and infralimbic cortices in response conflict affects behavioral flexibility in rats trained in a new automated strategy-switching task". In: *Behavioral Neuroscience* 123.5 (2009), pp. 979–991. DOI: 10.1037/a0016663.

[176] Zineb Ouhaz, Bethany A. L. Perry, Koji Nakamura, and Alexander S. Mitchell. "Mediodorsal Thalamus Is Critical for Updating during Extradimensional Shifts But Not Reversals in the Attentional Set-Shifting Task". In: *eNeuro* 9.2 (2022), ENEURO.0162–21.2022. DOI: 10.1523/ENEURO.0162-21.2022.

[177] Liam Paninski and John P. Cunningham. "Neural data science: Accelerating the experiment-analysis-theory cycle in large-scale neuroscience". In: *Current Opinion in Neurobiology* 50 (2018), pp. 232–241. DOI: 10.1016/j.conb.2018.04.007.

[178] Razvan Pascanu and Herbert Jaeger. "A Neurodynamical Model for Working Memory". In: *Neural Networks* 24.2 (2011), pp. 199–207. DOI: 10.1016/j.neunet.2010.10.003.

[179] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the Difficulty of Training Recurrent Neural Networks". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1310–1318.

[180] L.M. Peeters, R. Hinz, J.R. Detrez, S. Missault, W.H. De Vos, M. Verhoye, et al. "Chemogenetic silencing of neurons in the mouse anterior cingulate area modulates neuronal activity and functional connectivity". In: *NeuroImage* 220 (2020), p. 117088. DOI: 10.1016/j.neuroimage.2020.117088.

[181] Lawrence Perko. *Differential equations and dynamical systems*. Vol. 7. Springer Science & Business Media, 2013.

[182] Henri Poincaré. "Mémoire sur les courbes définies par une équation différentielle". French. In: *Journal de Mathématiques Pures et Appliquées* 7 (1881), pp. 375–422.

[183] Nathaniel James Powell and A. David Redish. "Representational changes of latent strategies in rat medial prefrontal cortex precede changes in behaviour". In: *Nature Communications* 7 (2016), p. 12830. DOI: 10.1038/ncomms12830.

[184] Gregory J. Quirk and Devin Mueller. "Neural mechanisms of extinction learning and retrieval". In: *Neuropsychopharmacology* 33.1 (2008), pp. 56–72. DOI: 10.1038/sj.npp.1301555.

[185] Michael E. Ragozzino, Susan Detrick, and Raymond P. Kesner. "Involvement of the prelimbic–infralimbic areas of the rodent prefrontal cortex in behavioral flexibility for place and response learning". In: *Journal of Neuroscience* 19.11 (1999), pp. 4585–4594. DOI: 10.1523/JNEUROSCI.19-11-04585.1999.

[186] Michael E. Ragozzino, Susan J. Mohler, Kenneth R. Prior, and Raymond P. Kesner. "The role of the dorsomedial striatum in behavioral flexibility for response and visual cue discrimination learning". In: *Behavioral Neuroscience* 117.5 (2003), pp. 1052–1063. DOI: 10.1037/0735-7044.117.5.1052.

[187] Rishi Rajalingham, Aída Piccato, and Mehrdad Jazayeri. "Recurrent Neural Networks with Explicit Representation of Dynamic Latent Variables Can Mimic Behavioral Patterns in a Physical Inference Task". In: *Nature Communications* 13.1 (2022), p. 5865. DOI: 10.1038/s41467-022-33581-6.

[188] Kanaka Rajan, Christopher D. Harvey, and David W. Tank. "Recurrent Network Models of Sequence Generation and Memory". In: *Neuron* 90.1 (2016), pp. 128–142. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2016.02.009.

[189] Richard H. Rand, Arthur H. Cohen, and Philip J. Holmes. "Systems of coupled oscillators as models of central pattern generators". In: *Neural Control of Rhythmic Movements in Vertebrates*. Ed. by Arthur H. Cohen. New York: Wiley, 1988, pp. 333–367.

[190] A. David Redish, Adam N. Elga, and David S. Touretzky. "A coupled attractor model of the rodent head direction system". In: *Network: Computation in Neural Systems* 7.4 (1996), pp. 671–685. DOI: 10.1088/0954-898X_7_4_004.

[191] Evan D. Remington, Devika Narain, Eghbal A. Hosseini, and Mehrdad Jazayeri. "Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics". In: *Neuron* 98.5 (2018), 1005–1019.e5. DOI: 10.1016/j.neuron.2018.05.020.

[192] Alfonso Renart and Christian K. Machens. "Variability in Neural Activity and Behavior". In: *Current Opinion in Neurobiology* 25 (2014), pp. 211–220. DOI: 10.1016/j.conb.2014.02.013.

[193] Erik L. Rich and Matthew Shapiro. "Rat prefrontal cortical neurons selectively code strategy switches". In: *Journal of Neuroscience* 29 (2009), pp. 7208–7219. DOI: 10.1523/JNEUROSCI.6068-08.2009.

[194] Matteo Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. "The Importance of Mixed Selectivity in Complex Cognitive Tasks". In: *Nature* 497.7451 (2013), pp. 585–590. DOI: 10.1038/nature12160.

[195] John Rinzel and G Bard Ermentrout. "Analysis of neural excitability and oscillations". In: *Methods in neuronal modeling* 2 (1998), pp. 251–292.

[196] John Rinzel and G. Bard Ermentrout. "Analysis of Neural Excitability and Oscillations". In: *Methods of Neuronal Modeling: From Synapses to Networks*. Ed. by Christof Koch and Idan Segev. Cambridge, MA: MIT Press, 1998, pp. 251–292.

[197] G. S. Robertson, S. R. Vincent, and H. C. Fibiger. "D1 and D2 dopamine receptors differentially regulate c-fos expression in striatonigral and striatopallidal neurons". In: *Neuroscience* 49.2 (1992), pp. 285–296. DOI: 10.1016/0306-4522(92)90096-K.

[198] T.J.M. Roelofs, J.P.H. Verharen, G.A.F. van Tilborg, L. Boekhoudt, A. van der Toorn, J.W. de Jong, et al. "A novel approach to map induced activation of neuronal networks using chemogenetics and functional neuroimaging in rats: A proof-of-concept study on the mesocorticolimbic system". In: *NeuroImage* 156 (2017), pp. 109–118. DOI: 10.1016/j.neuroimage.2017.05.008.

[199] Edmund T. Rolls, Lisa L. Critchley, Ursula V. Browning, and Frances M. Inoue. "Orbitofrontal cortex neurons: Role in olfactory and visual association learning". In: *Journal of Neurophysiology* 75.5 (1996), pp. 1970–1981. DOI: 10.1152/jn.1996.75.5.1970.

[200] F. J. Romeiras, Celso Grebogi, and Edward Ott. "Multifractal Properties of Snapshot Attractors of Random Maps". In: *Physical Review A* 41.2 (1990), pp. 784–799. ISSN: 1050-2947. DOI: 10.1103/PhysRevA.41.784.

[201] Cyrille Rossant et al. "Spike sorting for large, dense electrode arrays". In: *Nature Neuroscience* 19.4 (2016), pp. 634–641. DOI: 10.1038/nn.4268.

[202] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[203] Matthew F. S. Rushworth, Mark J. Buckley, Timothy E. J. Behrens, Mark E. Walton, and David M. Bannerman. "Functional organization of the medial frontal cortex". In: *Current Opinion in Neurobiology* 17.2 (2007), pp. 220–227. DOI: 10.1016/j.conb.2007.03.005.

[204] Eleonora Russo, Tianyang Ma, Rainer Spanagel, Daniel Durstewitz, Hazem Toutounji, and Georg Köhr. "Coordinated Prefrontal State Transition Leads Extinction of Reward-Seeking Behaviors". In: *Journal of Neuroscience* 41.11 (2021), pp. 2406–2419. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.2588-20.2021.

[205] Sadra Sadeh and Claudia Clopath. "Contribution of behavioural variability to representational drift". In: *eLife* 11 (2022), e77907. DOI: 10.7554/eLife.77907.

[206] 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". In: *Nature* 512.7515 (2014), pp. 423–426. DOI: 10.1038/nature13665.

[207] Lamia Jammal Salameh, Stephan H. Bitzenhofer, Ileana L. Hanganu-Opatz, Mathias Dutschmann, and Volker Egger. "Blood pressure pulsations modulate central neuronal activity via mechanosensitive ion channels". In: *Science* 383.6682 (2024), eadk8511. DOI: 10.1126/science.adk8511.

[208] Alexei Samsonovich and Bruce L. McNaughton. "Path Integration and Cognitive Mapping in a Continuous Attractor Neural Network Model". In: *The Journal of Neuroscience* 17.15 (1997), pp. 5900–5920. DOI: 10.1523/JNEUROSCI.17-15-05900.1997.

[209] Tim Sauer, James A. Yorke, and Martin Casdagli. "Embedology". In: *Journal of Statistical Physics* 65.3–4 (1991), pp. 579–616. DOI: 10.1007/BF01053745.

[210] Dominik Schmidt, Georgia Koppe, Zahra Monfared, Max Beutelspacher, and Daniel Durstewitz. "Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies". In: *International Conference on Learning Representations*. 2021.

[211] Geoffrey Schoenbaum, Matthew R. Roesch, Thomas A. Stalnaker, and Yuji K. Takahashi. "A New Perspective on the Role of the Orbitofrontal Cortex in Adaptive Behaviour". In: *Nature Reviews Neuroscience* 10.12 (2009), pp. 885–892. DOI: 10.1038/nrn2753.

[212] Martine R. van Schouwenburg, Marjolein P. Zwiers, Margot E. van der Schaaf, David E. M. Geurts, Arnt F. A. Schellekens, Jan K. Buitelaar, Robert J. Verkes, and Roshan Cools. "Frontostriatal involvement in task switching depends on genetic differences in D2 receptor density". In: *Journal of Neuroscience* 30.42 (2010), pp. 14205–14212. DOI: 10.1523/JNEUROSCI.1060-10.2010.

[213] Thomas Schreiber and Andreas Schmitz. "Improved Surrogate Data for Nonlinearity Tests". In: *Physical Review Letters* 77.4 (1996), pp. 635–638. DOI: 10.1103/PhysRevLett.77.635.

[214] Thomas Schreiber and Andreas Schmitz. "Surrogate Time Series". In: *Physica D: Nonlinear Phenomena* 142 (2000), pp. 346–382. DOI: 10.1016/S0167-2789(00)00043-9.

[215] Jeremy K. Seamans and Chen R. Yang. "The principal features and mechanisms of dopamine modulation in the prefrontal cortex". In: *Progress in Neurobiology* 74.1 (2004), pp. 1–58. DOI: 10.1016/j.pneurobio.2004.05.006.

[216] Susan R. Sesack, Susan W. King, Christine N. Bressler, Stanley J. Watson, and David A. Lewis. "Electron microscopic visualization of dopamine D2 receptors in the forebrain: Cellular, regional, and species comparisons". In: *Society for Neuroscience Abstracts*. Vol. 21. 1995, p. 365.

[217] H. Sebastian Seung. "How the Brain Keeps the Eyes Still". In: *Proceedings of the National Academy of Sciences of the United States of America* 93.23 (1996), pp. 13339–13344. DOI: 10.1073/pnas.93.23.13339.

[218] H. Sebastian Seung, Daniel D. Lee, Ben Y. Reis, and David W. Tank. "Stability of the Memory of Eye Position in a Recurrent Network of Conductance-Based Model Neurons". In: *Neuron* 26.1 (2000), pp. 259–271. DOI: 10.1016/S0896-6273(00)81155-1.

[219] Denis Sheynikhovich, Satoru Otani, Jing Bai, and Angelo Arleo. "Long-term memory, synaptic plasticity and dopamine in rodent medial prefrontal cortex: Role in executive functions". In: *Frontiers in Behavioral Neuroscience* 16 (2023), p. 1068271. DOI: 10.3389/fnbeh.2022.1068271.

[220] Hava T. Siegelmann and Eduardo D. Sontag. "On the Computational Power of Neural Nets". In: *Journal of Computer and System Sciences* 50.1 (1995), pp. 132–150. DOI: 10.1006/jcss.1995.1013.

[221] William E. Skaggs, James J. Knierim, Hemant S. Kudrimoti, and Bruce L. McNaughton. "A Model of the Neural Basis of the Rat's Sense of Direction". In: *Advances in Neural Information Processing Systems 7*. MIT Press, 1995, pp. 173–180.

[222] Hanlin F. Song, Guangyu R. Yang, and Xiao-Jing Wang. "Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework". In: *PLoS Computational Biology* 12.2 (2016), e1004792. DOI: 10.1371/journal.pcbi.1004792.

[223] Pengcheng Song and Xiao-Jing Wang. "Angular Path Integration by Moving 'Hill of Activity': A Spiking Neuron Model without Recurrent Excitation of the Head-Direction System". In: *The Journal of Neuroscience* 25.4 (2005), pp. 1002–1014. DOI: 10.1523/JNEUROSCI.4172-04.2005.

[224] Eduardo D. Sontag. "A Learning Result for Continuous-Time Recurrent Neural Networks". In: *Systems & Control Letters* 34.3 (1998), pp. 151–158. DOI: 10.1016/S0167-6911(98)00006-1.

[225] Timothy Spellman, Malka Svei, Jesse Kaminsky, Gabriela Manzano-Nieves, and Conor Liston. "Prefrontal deep projection neurons enable cognitive flexibility via persistent feedback monitoring". In: *Cell* 184.10 (2021), 2750–2766.e17. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2021.03.047.

[226] Michael R. Stefani, Katherine Groth, and Bita Moghaddam. "Glutamate receptors in the rat medial prefrontal cortex regulate set-shifting ability". In: *Behavioral Neuroscience* 117.4 (2003), pp. 728–737.

[227] Moira L. Steyn-Ross, D. Alistair Steyn-Ross, and Jamie W. Sleigh. "Chaotic dynamics underpins the slow oscillation of general anesthesia and nonREM sleep". In: *BMC Neuroscience* 13.Suppl 1 (2012), F3. DOI: 10.1186/1471-2202-13-S1-F3.

[228] Andrea Stocco, Christian Lebiere, and John R. Anderson. "Conditional Routing of Information to the Neocortex: A Network Model of Basal Ganglia Function". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 30. 2008, pp. 234–239.

[229] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. 2nd ed. Boulder, CO: Westview Press, 2015. ISBN: 978-0813349107.

[230] D. T. Stuss, G. P. Levine, M. P. Hamer, R. J. Palumbo, and F. Picton. "Relationship between frontal lobe lesions and Wisconsin Card Sorting Test performance in patients with multiple sclerosis". In: *Neuropsychology* 8.1 (1994), pp. 96–102.

[231] David Sussillo and Omri Barak. "Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks". In: *Neural Computation* 25.3 (2013), pp. 626–649. DOI: 10.1162/NECO_a_00409.

[232] Richard S. Sutton and Andrew G. Barto. "Reinforcement learning". In: *Journal of Cognitive Neuroscience* 11.1 (1999), pp. 126–134. DOI: 10.1162/089892999563265.

[233] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998. ISBN: 978-0-262-19398-6.

[234] A. Szelényi, T. K. Kracht, T. Meyer, J. Lange, P. Herholz, and V. P. Schramm. "Verbal fluency, Trail Making, and Wisconsin Card Sorting Test performance following right frontal lobe tumor resection". In: *Journal of Neurosurgery* 108.1 (2008), pp. 154–158.

[235] Floris Takens. "Detecting Strange Attractors in Turbulence". In: *Dynamical Systems and Turbulence, Warwick 1980*. Ed. by David Rand and Lai-Sang Young. Vol. 898. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 366–381. DOI: 10.1007/BFb0091924.

[236] Alfred Tarski. "Der Wahrheitsbegriff in den formalisierten Sprachen". In: *Studia Philosophica* 1 (1935), pp. 261–405.

[237] Gerald Teschl. *Ordinary Differential Equations and Dynamical Systems*. American Mathematical Society, 2012. ISBN: 978-0-8218-8328-0.

[238] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[239] Nelson K. B. Totah, Young B. Kim, Hojjatollah Homayoun, and Bita Moghaddam. "Anterior cingulate neurons represent errors and preparatory attention within the same behavioral sequence". In: *Journal of Neuroscience* 29 (2009), pp. 6418–6426.

[240] Hazem Toutounji and Daniel Durstewitz. "Detecting Multiple Change Points Using Adaptive Regression Splines With Application to Neural Recordings". In: *Frontiers in Neuroinformatics* 12 (2018). ISSN: 1662-5196. DOI: 10.3389/fninf.2018.00067.

[241] Joe Z. Tsien. "Linking Hebb's coincidence-detection to memory formation". In: *Current Opinion in Neurobiology* 10.2 (2000), pp. 266–273. DOI: 10.1016/S0959-4388(00)00078-6.

[242] H. B. M. Uylings, H. J. Groenewegen, and Bryan Kolb. "Do rats have a prefrontal cortex?" In: *Behavioural Brain Research* 146.1–2 (2003), pp. 3–17. DOI: 10.1016/j.bbr.2003.09.028.

[243] Robin R. Vallacher and Andrzej Nowak, eds. *Dynamical Systems in Social Psychology*. San Diego, CA: Academic Press, 1994.

[244] Vladimir Vapnik. "Principles of Risk Minimization for Learning Theory". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 4. Morgan Kaufmann, 1992, pp. 831–838.

[245] Ryan Vogt, Maximilian Puelma Touzel, Eli Shlizerman, and Guillaume Lajoie. "On Lyapunov Exponents for RNNs: Understanding Information Propagation Using Dynamical Systems Tools". In: *Frontiers in Applied Mathematics and Statistics* 8 (2022), p. 818799. DOI: 10.3389/fams.2022.818799.

[246] Saurabh Vyas, Matthew D. Golub, David Sussillo, and Krishna V. Shenoy. "Computation Through Neural Population Dynamics". In: *Annual Review of Neuroscience* 43.Volume 43, 2020 (2020), pp. 249–275. ISSN: 1545-4126. DOI: https://doi.org/10.1146/annurev-neuro-092619-094115.

[247] Jing Wang, Devika Narain, Eghbal A. Hosseini, and Mehrdad Jazayeri. "Flexible Timing by Temporal Scaling of Cortical Responses". In: *Nature Neuroscience* 21.1 (2018), pp. 102–110. DOI: 10.1038/s41593-017-0028-6.

[248] Maxwell B. Wang, Max G'Sell, James F. Castellano, R. Mark Richardson, and Avniel Singh Ghuman. "A Week in the Life of the Human Brain: Stable States Punctuated by Chaotic Transitions". In: *Research Square* (2023). Preprint. DOI: 10.21203/rs.3.rs-2752903/v1.

[249] Xiao-Jing Wang. "Decision making in recurrent neuronal circuits". In: *Neuron* 60.2 (2008), pp. 215–234. DOI: 10.1016/j.neuron.2008.09.034.

[250] Xiao-Jing Wang. "Probabilistic Decision Making by Slow Reverberation in Cortical Circuits". In: *Neuron* 36.5 (2002), pp. 955–968. DOI: 10.1016/S0896-6273(02)01092-9.

[251] Zhijie Wang and Hong Fan. "Dynamics of a Continuous-Valued Discrete-Time Hopfield Neural Network with Synaptic Depression". In: *Neurocomputing* 71.1–3 (2007), pp. 181–190. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2007.01.004.

[252] Christopher J. C. H. Watkins and Peter Dayan. "Q-learning". In: *Machine Learning* 8 (1992), pp. 279–292. DOI: 10.1007/BF00992698.

[253] Paul J. Werbos. "Backpropagation Through Time: What It Does and How to Do It". In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560. DOI: 10.1109/5.58337.

[254] Hassler Whitney. "Differentiable Manifolds". In: *Annals of Mathematics* 37.3 (1936), pp. 645–680. DOI: 10.2307/1968482.

[255] E. T. Whittaker. "On a New Method of Graduation". In: *Proceedings of the Edinburgh Mathematical Society* 41 (1922), pp. 63–75. DOI: 10.1017/S0013091500077853.

[256] Marco A. Wiering and Martijn van Otterlo. "Reinforcement Learning". In: *Reinforcement Learning: State-of-the-Art*. Ed. by Marco A. Wiering and Martijn van Otterlo. Vol. 12. Adaptation, Learning, and Optimization. Springer, 2012, pp. 3–42. DOI: 10.1007/978-3-642-27645-3_1.

[257] Ronald J. Williams and David Zipser. "Experimental Analysis of the Real-Time Recurrent Learning Algorithm". In: *Connection Science* 1.1 (1989), pp. 87–111. DOI: 10.1080/09540098908915631.

[258] Hugh R Wilson and Jack D Cowan. "Excitatory and inhibitory interactions in localized populations of model neurons". In: *Biophysical journal* 12.1 (1972), pp. 1–24.

[259] Hugh R. Wilson. *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience*. Oxford: Oxford University Press, 1999.

[260] Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. Trans. by C.K. Ogden. London: Kegan Paul, Trench, Trubner & Co., Ltd., 1922.

[261] Kong-Fatt Wong and Xiao-Jing Wang. "A recurrent network mechanism of time integration in perceptual decisions". In: *Journal of Neuroscience* 26.4 (2006), pp. 1314–1328. DOI: 10.1523/JNEUROSCI.3733-05.2006.

[262] Simon N Wood. "Statistical inference for noisy nonlinear ecological dynamic systems". In: *Nature* 466.7310 (2010), pp. 1102–1104. DOI: 10.1038/nature09319.

[263] Min Yan, Can Huang, Peter Bienstman, Peter Tino, Wei Lin, and Jie Sun. "Emerging Opportunities and Challenges for the Future of Reservoir Computing". In: *Nature Communications* 15.1 (2024), p. 2056. DOI: 10.1038/s41467-024-45187-1.

[264] Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. "Task Representations in Neural Networks Trained to Perform Many Cognitive Tasks". In: *Nature Neuroscience* 22.2 (2019), pp. 297–306. DOI: 10.1038/s41593-018-0310-2.

[265] Robert S. Zucker and Wade G. Regehr. "Short-term synaptic plasticity". In: *Annual Review of Physiology* 64.1 (2002), pp. 355–405. DOI: 10.1146/annurev.physiol.64.092501.114547.