Dissertation

submitted to the

Combined Faculty of Mathematics, Engineering and Natural Sciences

of Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Manuel Benjamin Brenner

born in: Bietigheim-Bissingen

Oral examination: 03.07.2024

Learning Interpretable Dynamical Systems Models from Multimodal Empirical Time Series

Referees: Prof. Dr. Daniel Durstewitz

Prof. Dr. Tristan Bereau

ABSTRACT

Dynamical systems (DS) theory provides a rich framework to model dynamic processes across science and engineering. However, traditional scientific model building is often laborious and struggles with the complexities of real-world DS. Advances in machine learning (ML) have led to the development of automated, data-driven techniques for approximating governing equations from time series, called Dynamical Systems Reconstruction (DSR). Yet, these approaches often struggle with real-world systems characterized by chaos, noise, non-Gaussian and multimodal observations, or multistability. The black-box nature of many ML models further complicates their analysis even if they describe the data well. This thesis introduces novel methods for inferring interpretable DSR models from challenging empirical time series. This includes several recurrent neural network models and training algorithms, tailored to extracting low-dimensional and tractable DSR models, and a flexible framework for DSR from multimodal and non-Gaussian observations. It further introduces a hierarchical inference framework, an analysis pipeline for a class of piecewise linear DSR models, and a novel pruning approach that yields interpretable network topologies. Extensive comparisons to state-of-the-art DSR algorithms illustrate the significant advancements made by the proposed methods, promising applications in physics, neuroscience, and beyond.

ZUSAMMENFASSUNG

Die Theorie dynamischer Systeme (DS) bietet einen reichen Rahmen für die Modellierung dynamischer Prozesse in Wissenschaft und Technik. Die traditionelle wissenschaftliche Modellbildung ist jedoch oft mühsam und hat Schwierigkeiten, die Komplexität realer DS abzubilden. Fortschritte im maschinellen Lernen (ML) haben zur Entwicklung automatisierter, datengesteuerter Verfahren für die Approximation zugrundeliegender Gleichungen aus Zeitreihen geführt, die als dynamische Systemrekonstruktion (DSR) bezeichnet werden. Diese Ansätze haben jedoch oft Probleme mit realen Daten, die durch Chaos, Rauschen, nichtgaußsche und multimodale Beobachtungen oder Multistabilität gekennzeichnet sind. Die Black-Box-Natur vieler ML-Modelle macht ihre Analyse schwierig, selbst wenn sie die Daten gut beschreiben. In dieser Arbeit werden neue Methoden zur Ableitung interpretierbarer DSR-Modelle aus empirischen Zeitreihen vorgestellt. Dazu gehören mehrere rekurrente neuronale Netzwerkmodelle und Trainingsalgorithmen, die auf die Extraktion niedrigdimensionaler und interpretierbarer DSR-Modelle zugeschnitten sind, sowie ein flexibler Ansatz für DSR aus multimodalen und nicht-gaußschen Beobachtungen. Darüber hinaus werden ein hierarchischer Inferenzansatz, eine Analysemethode für eine Klasse von stückweise linearen DSR-Modellen und ein neuartiger Pruning-Ansatz vorgestellt, der interpretierbare Netzwerktopologien liefert. Ausführliche Vergleiche mit führenden DSR-Algorithmen veranschaulichen die bedeutenden Fortschritte der vorgestellten Methoden, die vielversprechende Anwendungen in Physik, Neurowissenschaften und darüber hinaus bieten.

Work is the place where the self meets the world.

— David Whyte

ACKNOWLEDGMENTS

A PhD is a time of ups and downs, of excitement and struggle. My PhD began only weeks before the onset of a global pandemic, shared some of his milestones with the outbreak of war in Europe, and was accompanied by an unprecedented hype around AI. Needless to say, it was an exciting, sometimes challenging time, but I am very grateful for all the things discovered, learned, and shared, and all the support I received.

Thanks to Daniel, for teaching me to always approach science with curiosity, passion, and integrity, no matter how stressful and frustrating it might get. Thanks to Georgia, for braving the perhaps biggest challenge of my PhD, the bureaucracy of an EU project, together, and for the many encouragements over the years.

Thanks goes to my family, for always being the rock that everything else is built on, for being patient hosts in times of crisis and cheerleaders in times of accomplishment. Thanks to Darshana, for sharing much of this PhD journey with me, for her talent being immortalized in my first Figure 1.

Thanks to Flo, for infusing the slings and arrows of constant deadlines with a sense of gallows humor, and being the partner-in-crime in the orders of magnitude less glorious Lennon/McCartney of dynamical systems reconstruction. To Leo, for bringing me into this group and introducing me into the world of dynamical systems. To Max, Janik, Lukas, Jonas, Zahra, Niclas, Daniel Kramer, and many other colleagues, for being friends on top of colleagues, and for making the office a cherished place of laughter.

Many thanks to the proofreaders Flo, Darshana, Sven, and Hanna, and thanks to Alex for his invaluable input on dynamical systems in quantum physics.

Dedicated to the loving memory of Hildegard Wolfram.
1926 – 2024

PUBLICATIONS

This PhD thesis is primarily based on the ideas and content of the following publications and preprints.

[1] Manuel Brenner, Christoph Hemmer, and Daniel Durstewitz. "Almost-Linear RNNs Yield Highly Interpretable Symbolic Codes in Dynamical Systems Reconstruction." Preprint. 2024.

[2] Manuel Brenner, Florian Hess, Georgia Koppe, and Daniel Durstewitz. *Integrating Multimodal Data for Joint Generative Modeling of Complex Dynamics*. Published at AAAI 2023 MLmDS workshop as "Multimodal Teacher Forcing for Reconstructing Nonlinear Dynamical Systems". Accepted at ICML 2024. DOI: 10.48550/arXiv.2212.07892.

[3] 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*. ISSN: 2640-3498. PMLR, June 2022, pp. 2292–2320. URL: https://proceedings.mlr.press/v162/brenner22a.html.

[4] Manuel Brenner, Christoph Korn, Daniela Mier, Stephanie N.L. Schmidt, Stefanie Lis, Peter Kirsch, Daniel Durstewitz, and Georgia Koppe. "Creating digital twins of social interaction partners through deep dynamical systems learning." Preprint. 2024.

[5] Niclas Göring, Florian Hess, Manuel Brenner, Zahra Monfared, and Daniel Durstewitz. *Out-of-Domain Generalization in Dynamical Systems Reconstruction*. Accepted at ICML 2024. DOI: 10.48550/arXiv.2402.18377.

[6] Christoph Hemmer, Manuel Brenner, Florian Hess, and Daniel Durstewitz. "Optimal Network Topologies for Dynamical Systems Reconstruction." Accepted at ICML 2024.

[7] 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*. ISSN: 2640-3498. PMLR, July 2023, pp. 13017–13049. URL: https://proceedings.mlr.press/v202/hess23a.html.

CONTENTS

I INTRODUCTION 1

1 SUMMARY 3
- 1.1 Main Contributions 6
  - 1.1.1 Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems 6
  - 1.1.2 Generalized Teacher Forcing for Learning Chaotic Dynamics 7
  - 1.1.3 Integrating Multimodal Data for Joint Generative Modeling of Complex Dynamics 7
  - 1.1.4 Creating Digital Twins of Social Interaction Partners through Deep Dynamical Systems Learning 8
  - 1.1.5 Out-of-Domain Generalization in Dynamical Systems Reconstruction 9
  - 1.1.6 Optimal Network Topologies for Dynamical Systems Reconstruction 10
  - 1.1.7 Hierarchical Inference Framework 10
  - 1.1.8 Analyzing Linear Subregions of Inferred RNNs 11

2 INTRODUCTION 13
- 2.1 Introduction 13
- 2.2 Dynamical Systems 15
- 2.3 Dynamical Systems Reconstruction (DSR) 18
- 2.4 Machine Learning Techniques for DSR 23
- 2.5 Generalisation in DSR 26
- 2.6 Hierarchisation and Transfer Learning in DSR 28
- 2.7 Applications of DSR 29

II METHODS 33

3 NETWORK MODELS AND TRAINING ALGORITHMS 35
- 3.1 RNN Models 35
  - 3.1.1 Dendritic PLRNN 37
  - 3.1.2 Shallow PLRNN 38
- 3.2 Training Algorithms 40
  - 3.2.1 Challenges of Training RNN Models on Chaotic Dynamical Systems 40
  - 3.2.2 Sequential Variational Autoencoders (SVAEs) 42
  - 3.2.3 Sparse Teacher Forcing (STF) 46
  - 3.2.4 Generalized Teacher Forcing (GTF) 48
  - 3.2.5 Multimodal Teacher Forcing (MTF) 50
  - 3.2.6 Encoder Models 56
  - 3.2.7 Decoder Models 58
- 3.3 Hierarchization Framework 60
  - 3.3.1 Hierarchical shPLRNN 60

III RESULTS 63

4 RESULTS 65

4.1 Performance Metrics in DSR 65
4.1.1 Performance Metrics for Unimodal Continuous Time Series 66
4.1.2 Performance Metrics for Multimodal Time Series 69
4.1.3 Generalization Error in DSR 72
4.2 Reconstructions from Unimodal Time Series 74
4.2.1 Unimodal Benchmarks 74
4.2.2 Unimodal Experimental Results 78
4.3 Reconstructions from Discrete and Multimodal Time Series 80
4.3.1 Multimodal Benchmarks 80
4.3.2 DSR from Discrete Random Variables 85
4.3.3 Multimodal Experimental Results 88
4.4 Hierarchisation Framework 93
4.4.1 Benchmark Evaluation 93
4.4.2 Applications to fMRI data 93
4.5 Analysis of Linear Subregions of Trained PLRNNs 95
4.6 Optimal Network Topologies for Dynamical Systems Reconstruction 101
5 DATA-DRIVEN LEARNING OF SOCIAL INTERACTION DYNAMICS 105
5.1 Introduction 105
5.2 Analysis of State Space of Inferred Models 109
5.3 Model Simulations 112
5.4 Interpretation 116
6 CONCLUSIONS AND OUTLOOK 119
6.1 Conclusions 119
6.2 Limitations and Outlook 121
IV APPENDIX 127
A APPENDIX METHODS 129
A.1 Training Details 129
A.2 Comparison Methods 130
A.2.1 Unimodal Comparisons 130
A.2.2 Multimodal Comparisons 136
A.3 Datasets 137
A.3.1 Benchmark Systems 137
A.3.2 Experimental Datasets 139
B APPENDIX RESULTS 141
B.1 Further Results 141
List of Figures 145
BIBLIOGRAPHY 156

ACRONYMS

AI	Artificial Intelligence
BPTT	Backpropagation Through Time
CNN	Convolutional Neural Network
DSR	Dynamical Systems Reconstruction
ECG	Electrocardiogram
EEG	Electroencephalography
ELBO	Evidence Lower Bound
EVGP	Exploding and Vanishing Gradient Problem
fMRI	functional Magnetic Resonance Imaging
KL	Kullback–Leibler
LSTM	Long Short Term Memory
ML	Machine Learning
ODE	Ordinary Differential Equation
RC	Reservoir Computing
RNN	Recurrent Neural Network
SOTA	State-of-the-Art
SSM	State-Space Model
TF	Teacher Forcing
STF	Sparse Teacher Forcing
GTF	Generalized Teacher Forcing
MTF	Multimodal Teacher Forcing
TG	Trust Game
TSF	Time Series Forecasting

Part I

INTRODUCTION

1

SUMMARY

The discovery of governing equations from empirical data has underpinned scientific progress for centuries. Dynamical systems (DS) theory provides a powerful mathematical framework to describe dynamical processes and their properties in the language of differential equations and discrete maps. DS models are prevalent in many scientific areas, from physics, neuroscience, and climate science to finance and engineering. A good DS model that reflects the underlying dynamical processes well can give rise to insights about the system it describes, from capturing its sensitivity to perturbation, its bifurcations, or, in the case of neuroscience, the computational mechanism implemented by the neural dynamics.

However, in many scientific contexts, we often begin with noisy time series observations without a clear understanding of the DS underlying them. Especially for complex DS, constructing models from the ground up may not fully capture the dynamics' complexity, may introduce the scientist's biases into the model, and can be very time-consuming. Consequently, learning-based approaches have gained popularity for the automated approximation of governing equations from time series observations, known as Dynamical Systems Reconstruction (DSR). With the recent success of Machine Learning (ML) and Artificial Intelligence (AI) models in various scientific and applied areas, most leading DSR approaches are based on ML techniques. Although the ultimate goal of DSR algorithms is to approximate DS models from experimental datasets, the focus in the field has predominantly been on reconstructing benchmark systems derived from synthetic and often relatively simple models. However, these benchmarks frequently fail to address several practical challenges inherent in real-world data:

- the ubiquity of chaos in real-world systems
- noisy and partial observations
- non-Gaussian measurements that significantly coarse-grain or alter the representation of the underlying DS, such as discrete ordinal scales in psychology, count data in neuroscience, or non-Gaussian Levy processes in turbulent fluid flows
- multiple jointly observed data channels, combining Gaussian and non-Gaussian observations
- the presence of multiple dynamical regimes in the same underlying system (multistability)
- short experimental time series, observed across multiple related systems or subjects

Automated regression-based approaches following in the footsteps of classical scientific model building largely fail on real-world data due to these challenges. Meanwhile, ML models based on universal approximators are often

black-box in nature, making it challenging to analyze models even if they have been successfully inferred.

This thesis's primary contributions, detailed further in Chapter 1.1, address all these challenges, introducing models and training algorithms equipped to extract interpretable and tractable DSR models from challenging empirical time series data.

Chapter 2 provides an overview of DS theory and DSR methods, with a particular emphasis on ML-based approaches. It also explores the current frontiers in DSR, including multimodality, out-of-domain generalization, and transfer learning, and outlines applications of DSR in physics, neuroscience, and psychiatry.

Chapter 3 presents the DSR models and training algorithms used throughout the thesis. Sect. 3.1 introduces two recurrent neural network (RNN) models, the dendritic and shallow piecewise linear (PL) RNN, that facilitate DSR in low-dimensional latent spaces while maintaining the mathematical tractability of the standard PLRNN. Chapter 3.2 presents training algorithms based on sequential variational autoencoders (SVAEs) and sparse and generalized teacher forcing (STF/GTF), tailored for DSR from time series observations of chaotic DS. The multimodal teacher forcing (MTF) technique combines all methods into a comprehensive probabilistic DSR framework for inferring DSR models from discrete or multimodal time series. The hierarchical inference framework (Sect. 3.3) enables DSR from multiple time series, such as measurements from different but related physical systems or multiple subjects in a clinical context. It enables transfer learning by combining group-level and individual-level information while extracting interpretable low-dimensional substructure.

Chapter 4 demonstrates DSR on a wide range of benchmark and empirical time series and extensive comparisons to many other state-of-the-art (SOTA) DSR algorithms. Sect. 4.1 first introduces a range of performance metrics for assessing DSR quality, based on invariant geometric, temporal, and topological properties of reconstructed systems, including modality-specific metrics. Sect. 4.1.3 introduces a generalization error for DSR, which is used to define the concept of the learnability of an out-of-domain generalization problem in DSR. Sect. 4.2.1 showcases DSR using STF and GTF from noisy, short, and partially observed time series, and from several experimental time series (Sect. 4.2.2), on which other SOTA approaches fail. Sect. 4.3.1 provides reconstructions from multimodal benchmarks using MTF, while Sect. 4.3.2 showcases DSR solely from discrete symbolic and ordinal encodings of chaotic DS. Sect. 4.3.3 finally showcases reconstructions from multimodal experimental time series, illustrating the benefits of multimodal data integration in empirical settings. Sect. 4.4 presents results using the hierarchization framework, extracting interpretable low-dimensional structure from benchmark systems and learning a joint DSR model across experimental time series measured from different subjects. Sect. 4.5 illustrates results interpreting inferred PLRNN models with respect to the linear subregions they inhabit, extracting low-dimensional sparse graph representations from several benchmark systems. Sect. 4.6 summarizes key results using a novel pruning approach tailored to DSR, removing weights based on their influence on attractor agreement, which enables reconstructions with sparser models and the extraction of interpretable network topologies beneficial for DSR. Chapter 5 presents results using MTF on short, ordinal experimental time series within a psychological trust game paradigm, creating 'digital twins' of

complex human social interactions. It illustrates several ways to interpret inferred models, leveraging their generative dynamics for novel scientific insight. Finally, Chapter 6 discusses limitations and future research directions.

The methods developed in this thesis, alongside the experimental results, mark a significant advancement in the field of DSR. This is underscored by extensive comparisons to other existing SOTA algorithms for DSR on a range of simulated and real-world datasets both in the unimodal and multimodal case (Sect. A.2), and quantified by a range of performance metrics specifically tailored to DSR (Sect. 4.1). The models are general and allow for a wide range of applications in all disciplines where time series data are measured, from physics to neuroscience and psychiatry, offering promise for future scientific research and clinical practice.

1.1 MAIN CONTRIBUTIONS

This section outlines the key contributions from the papers and preprints authored during my PhD. The first four papers summarized in the following (Brenner et al. [53–55] and Hess et al. [152]) form the core of this thesis. Concepts and results from Göring et al. [139] and Hemmer et al. [149] are discussed more briefly in various parts of this thesis. My own publications are indicated with colored brackets for easy reference. The linear subregions analysis approach is in preparation for submission to NeurIPS 2024 as part of Brenner, Hemmer, and Durstewitz [52]. The hierarchisation approach has not yet been published but will be transformed into a publication in the upcoming months.

For figures and tables derived from or taken from these publications, I have cited the respective papers. All figures displayed in this thesis, including those taken from or based on publications, were either solely or substantially created by me. For all methods and results from those papers, I have included only those to which I have made significant contributions, with the specific contributions for each paper outlined below.

1.1.1 *Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems*

PROBLEM Current ML models for DSR are often mathematically complex and intractable and require high-dimensional latent spaces to properly reconstruct DS from observed time series data. Other types of models require prior knowledge about the true system's functional form or are not dynamically universal.

SOLUTION Expanding on the framework of piecewise linear (PL) RNNs [95], in Brenner et al. [54] we introduce the dendritic PLRNN (dendPLRNN), a mathematically tractable and more expressive variant of the PLRNN, inspired by the principles of dendritic computation. This approach retains several of the benefits of the PLRNN, such as analytical access to system characteristics like fixed points and k-cycles, and it allows for the conversion into a continuous time model, simplifying dynamical system (DS) analysis post-training. At the same time, it allows reconstructions in significantly lower-dimensional state spaces by making the computations of the individual units of the PLRNN more expressive. We introduce two different training frameworks for the dendPLRNN, one based on sequential variational autoencoders (SVAEs), which integrate the dendPLRNN into a fully probabilistic training paradigm, and one based on backpropagation through time (BPTT), combined with sparse teacher forcing (STF), called BPTT-TF. An additional contribution of this study lies in assembling a collection of evaluation methods, which include short-term prediction errors and long-term invariant statistics of the reconstructed systems. These methods were used to benchmark a range of state-of-the-art (SOTA) DS algorithms. Training the dendPLRNN with BPTT-TF outperformed all other approaches, successfully reconstructing even challenging experimental datasets.

MY CONTRIBUTION As first author, I was involved in most aspects of the paper. Together with Leonard Bereska, I was mainly responsible for structuring the paper, searching for comparison methods and performance measures,

and compiling benchmark systems and experimental datasets. On the empirical side, I evaluated the results using the SVAE approach and several comparison methods (SINDy, Neural ODEs), while results with BPTT-TF and other comparisons were primarily contributed by Florian Hess. The formal proofs were mainly provided by Zahra Monfared and Jonas Mikhaeil.

1.1.2 *Generalized Teacher Forcing for Learning Chaotic Dynamics*

PROBLEM Training RNN models on chaotic dynamics is challenging because exponential trajectory divergence in chaotic systems is intimately tied to gradient divergence in BPTT-based training. On the other hand, long training sequences are beneficial for capturing the long-term behavior of the underlying system, particularly when training on experimental data.

SOLUTION After the success of BPTT-TF-based training in Brenner et al. [54], in Hess et al. [152] we further developed the idea of training RNNs with BPTT-TF into the generalized teacher forcing (GTF) framework. We show that this training method can ensure provably bounded gradients throughout training on chaotic systems. We propose an adaptive training framework where optimal forcing strengths are estimated and adjusted during training to optimally balance gradients. We further introduce a novel variant of the PLRNN, the 'shallow PLRNN' (shPLRNN), which, just as the dendPLRNN, retains the original PLRNN's mathematical tractability, while further decreasing the number of latent dimensions required for successful reconstructions, in many cases even achieving reconstructions in the observation dimension of the system. We show that the shPLRNN trained with GTF particularly excels at DSR from experimental time series, outperforming many other SOTA approaches in the field which overwhelmingly fail on real-world data.

MY CONTRIBUTION I was involved in developing the shPLRNN architecture, in conceptual discussions surrounding GTF and the adaptive annealing protocol, and in writing the paper. On the empirical side, I evaluated most of the comparison methods, including SINDy, LEM, Neural ODEs, ODE-RNN, and Latent ODE.

1.1.3 *Integrating Multimodal Data for Joint Generative Modeling of Complex Dynamics*

PROBLEM Most DS are empirically accessed via measurements. In many practical settings, these measurements can be discrete, such as survey data in psychology, or event counts in climate science, or only represent partial observations of the underlying system. Often, several different modalities are observed simultaneously, such as behavioral data and neural recordings in neuroscience, and often feature multimodal cross-modal links. While multimodality is by now increasingly popular in AI as a whole, in the field of DSR, efficient training algorithms to address non-Gaussian measurements and multimodality are essentially lacking.

SOLUTION In Brenner et al. [53], we propose a general training framework for DSR from multimodal and non-Gaussian data called Multimodal Teacher Forcing (MTF). This approach integrates TF-based training into a fully probabilistic framework capable of DSR from any combination of jointly observed time series, even when these follow different distributional assumptions. We demonstrate that the MTF approach significantly surpasses all other tested methods, including a multimodal sequential variational autoencoder [214], across a range of simulated and empirical benchmarks. We showcase that DSR remains feasible even in the presence of heavy distortion by observation noise through multimodal data integration. We illustrate that the MTF framework naturally handles missing observations, and learns temporal delay embeddings automatically from data when an underlying DS is only partially observed. We also show for the first time in the literature that DSR from a purely categorical (symbolic) representation of chaotic attractors is achievable. In addition, we compile a range of modality-specific and general performance metrics, introducing for instance a measure that compares reconstructed attractor geometry in cases in which continuous observations of the underlying system are lacking. Lastly, we illustrate the advantages of multimodal data integration through analysis of two experimental datasets: one incorporating functional magnetic resonance imaging (fMRI) and behavioral data, and another combining spike data from the rat hippocampus with simultaneously observed position data. These results reveal interesting cross-modal links in the reconstructed DSR models that the MTF approach can leverage.

MY CONTRIBUTION The MTF approach was developed primarily by me with some conceptual discussions with Florian Hess. The implementation of the approach and all empirical evaluations, including the compilation and evaluation of all comparison methods, benchmark datasets, and empirical datasets, were carried out by me.

1.1.4 *Creating Digital Twins of Social Interaction Partners through Deep Dynamical Systems Learning*

PROBLEM Traditional methods of studying social interactions, such as social exchange games in controlled environments, lead to complex and multi-faceted behavioral outcomes. One key challenge in modeling these lies in disentangling the computational mechanisms underlying these interactions and taking into account individual differences. Previous modeling approaches, such as those based on Reinforcement Learning (RL), have been theory-driven and hand-crafted, potentially missing out on the true generative mechanisms due to their reliance on the model builder's prior knowledge.

SOLUTION In Brenner et al. [55], we explore purely data-driven models, specifically dendPLRNNs trained with the MTF framework, as a novel approach to learning the generative computational mechanisms underlying social interactions. We demonstrate that these models can accurately predict out-of-sample behavior and replicate behavioral group statistics. We further find that the inferred state spaces of the RNNs effectively encode investment decisions and their associated uncertainty, and distinctly encoded external cues like facial

identity and emotional expressions. We also used trained models to simulate completely novel interactions, leveraging the generative nature of the algorithm. Here we discovered bifurcations between exploratory and stable investment behavior based on external cues, and interpretable clusters in the interaction styles across subjects when paired with novel trustees. Our findings demonstrate the capability of data-driven RNN models to decode and predict complex social interaction patterns, opening the door to applications in psychology and beyond.

MY CONTRIBUTION All analyses discussed in Chapter 5 were performed by me. The conceptual discussions and writing of the paper were carried out jointly with Georgia Koppe.

1.1.5 *Out-of-Domain Generalization in Dynamical Systems Reconstruction*

PROBLEM While the field of DSR has been increasingly successful in extracting dynamics models from data that share the same long-term statistics as the true underlying system, a general framework for studying out-of-domain generalization in DSR is still lacking. For instance, while many real-world systems, such as the brain or the climate, are believed to be multistable, the challenge of learning multistable DS is essentially unstudied in the literature.

SOLUTION In Göring et al. [139], we developed a formal framework that addresses generalization in DSR, highlighting the unique aspects of out-of-domain generalization (OODG) in DSR compared to other areas of ML. We demonstrate that there are close ties between OODG and multistability, and introduce concepts derived from topology and ergodic theory to define topological and statistical generalization errors, which are shown to be sensitive to multistability. Building on this understanding of generalization, we introduce the learnability of an OODG problem in DSR. We formally prove that a class of algorithms that create a strong prior by explicitly providing a function class for the underlying DS and are trained via linear parameter estimation can successfully generalize. We show that ML techniques without a similarly strong prior generally fail to learn a DSR model capable of effective generalization. We also empirically validate this by assessing several major classes of DSR algorithms, identifying where and why they fall short in generalizing across the entire phase space for several multistable benchmark systems. Our work presents the first comprehensive mathematical treatment of OODG in DSR, offering deeper insights into the core challenges of OODG and potential strategies to address them.

MY CONTRIBUTION I developed the conceptual outline of the paper jointly with Niclas Göring, particularly concerning the general framework for OODG in DSR and the learnability distribution (Sect. 2.5), and the framing of the experimental section with both Niclas Göring and Florian Hess. I also performed an extensive literature search, including the classification of benchmark systems provided in Table 1. Additionally, I performed all empirical analyses based on symbolic regression using SINDy (see Appx. A.2.1).

1.1.6 *Optimal Network Topologies for Dynamical Systems Reconstruction*

PROBLEM Small and parsimonious models with a small parameter count are often desirable in interpreting trained ML models. Pruning procedures that iteratively prune parameters based on their magnitude after starting with highly overparameterized networks have been particularly successful in recent years. In the context of feedforward NNs and tasks like image classification, they have uncovered 'lottery tickets' of subnetworks that are particularly effective at solving a given task with much fewer parameters. However, magnitude-based pruning approaches commonly used for other ML tasks fail when applied to DSR.

SOLUTION In Hemmer et al. [149] we introduce a new pruning procedure called 'geometric pruning' tailored for pruning of DSR models. This approach contrasts with traditional magnitude-based pruning by focusing on removing weights that have a low contribution to the reconstructed attractor's geometrical structure and hence is sensitive to how changes in the parameter affect the dynamics of the reconstructed system. This method can drastically reduce the parameter load of models without significantly affecting the quality of DSR, and results in sparse networks with a particular network topology composed of hubs and featuring small-world substructure. This resulting topology, rather than the magnitude of weights of the initialization, is crucial for improving DSR using the resulting networks. Inspired by this result, we reverse-engineer an algorithm that generates such topologies, which can be used as a prior for initializing new DSR models. Compared with other well-studied topologies like small-world or scale-free networks, this approach leads to faster convergence of trained models and highly sparse and interpretable models.

MY CONTRIBUTION My main contributions included conceptual discussions, contributions to the related work, especially on network topology in machine learning, and the writing and creation of figures for the paper (e.g. Fig. 39). The idea of disentangling network topology from the specific random network weights was developed jointly with Christoph Hemmer. Additionally, I tested other pruning procedures, such as pruning based on the Lyapunov spectrum, and applied these techniques to other network architectures, such as the sh-PLRNN.

1.1.7 *Hierarchical Inference Framework*

PROBLEM In experimental settings, we often record short time series across multiple subjects. While individual time series are expected to share characteristics with the group, facilitating transfer learning across subjects, the differences between subjects necessitate subject-specific fine-tuning of models. An optimal inference procedure should integrate data from different subjects, while also retaining individual differences and identifying latent interpretable structures encoding these individual differences.

SOLUTION Sect. 3.3 introduces a general and flexible inference framework for hierarchization and transfer learning in DSR. In this context, parameter hi-

erarchization involves varying levels of parameter inference, where the upper hierarchy parameters are inferred from time series data derived from multiple subjects, and the lower hierarchy from individual subjects' data. Specifically, we introduce a low-dimensional, subject-specific parameter vector designed to capture all relevant subject-specific differences. This vector is projected onto the parameters of a DSR model through projection matrices jointly trained across all subjects. After training, the extracted low-dimensional feature vector can be further analyzed, and related to ground-truth parameters in benchmark DS, or psychologically or clinically relevant differences between subjects for experimental time series.

MY CONTRIBUTION All results and analyses displayed in Sect. 4.4 were implemented and performed by me. The extraction of low-dimensional parameter vectors from the Lorenz-63 system was first tested jointly with Elias Weber.

1.1.8 *Analyzing Linear Subregions of Inferred RNNs*

PROBLEM DSR algorithms often employ universal approximators such as RNNs to model DS. Although these models can represent any DS, they are typically difficult to analyze, and the methods through which they approximate functions are not well understood. Specifically, piecewise linear RNNs (PLRNNs) use a network architecture that divides the RNN's state space into subregions with locally linear dynamics, representing nonlinear DS through transitions between these linear subregions. Despite a longstanding interest in using more analytically tractable linear models to capture complex nonlinear dynamics, the exact mechanisms by which PLRNNs achieve this are still not well understood.

SOLUTION Sect. 4.5 introduces a pipeline for analyzing trained PLRNN models in terms of the linear sub-regions reconstructed systems inhabit, and presents results on five benchmark systems. Although the number of available subregions increases exponentially with larger PLRNN models, reconstructed systems tend to occupy only a limited subset of these. Within this subset, the dynamics of the system are primarily implemented in an even smaller group of sub-regions that frequently transition between each other. The transitions between different linear subregions in this dominant subgroup can be represented as sparse graphs, exhibiting graph-theoretic properties such as small-world structures. These properties reflect the nature of the reconstructed DS, distinguishing, for example, between chaotic behavior and limit cycles.

MY CONTRIBUTION All results and analyses discussed in Sect. 4.5 were performed by me.

2 INTRODUCTION

All models are wrong, but some are useful.

— George Box

2.1 INTRODUCTION

At least since the advent of Greek philosophy, humans have recognized that the world is in constant flux. Heraclitus famously stated that 'panta rhei': everything flows, while in Buddhism, the concept of 'anicca' (impermanence) similarly enshrines the flowing nature of reality. For an equally long time, humans have tried to understand the nature of this flux. Millennia of inquiry have led to the development of a rich theoretical framework for describing the flowing world around us. Johannes Kepler spent four years analyzing the planetary orbits observed by Tycho Brahe, ultimately formulating his famous laws. The development of calculus by Newton and Leibniz introduced a formal language for expressing continuous change through infinite sums of infinitesimal changes. This advancement opened the door for many scientific breakthroughs and continues to underpin the language of science and engineering to this day [381].

The cycle of experimenters collecting data and theoreticians constructing models to explain this data has been fundamental to scientific progress for centuries. However, with the advent of machine learning (ML) and artificial intelligence (AI), the scientific discovery process is undergoing a significant transformation. The emerging 'culture' [51] of ML emphasizes algorithmic models that learn or discover data-generating mechanisms directly from data. Whereas Kepler spent four laborious years brute-forcing his way through equations to discover that ellipses described the planetary orbits well, 'AI Feynman' can derive these laws directly from data in seconds [401] by determining which function from a large library of candidate functions best describes the data.

ML research focuses on creating the right conditions for learning algorithms to succeed. Once these conditions are met, and given sufficient computational resources, model discovery occurs 'automatically'. This changes the way scientific research is conducted: new models are not formulated from scratch but by providing a blueprint and an optimization framework that enables algorithms to discover models by themselves. While not built from the bottom up to encode scientific knowledge or test hypotheses, models are often still designed to advance the scientific discovery process in ways that raw data cannot, such as by making out-of-sample predictions, extracting low-dimensional patterns from big data, or by allowing access to properties of the systems they describe that are otherwise unattainable.

ML has undergone a dramatic boom during the course of this thesis, especially in the space of generative modeling, with models such as DALL-E [323] or GPT-4 [292] taking the world by storm. State-of-the-art (SOTA) ML models are now trained with up to a factor of 10^{10} [383] of the computational resources

than only ten years ago when breakthroughs like AlexNet [216] brought deep learning (DL) to the forefront of AI research. The 'unreasonable effectiveness of deep learning in artificial intelligence' [360] now means that approaches based on DL, and associated large-scale models, are increasingly dominant in the fields of AI and ML. The advances in computational capacity and model size through the rise of DL even go far beyond the scaling of Moore's law, with its doubling of transistor density on a chip roughly every two years holding since the 1970s. The 'DL revolution' [359] has led to major advancements in AI capabilities, but it has also substantially increased the cost of training foundation models [292, 448], restricting some types of research to well-funded large tech companies or start-ups with significant backing.

Models are now often formulated as gigantic black boxes that learn patterns from equally gigantic amounts of data. Large language models (LLMs) perform well across a range of tasks [448], but how they achieve this is often still a mystery. Scientists are now leveraging tools like cognitive psychology [41] to study LLMs, much like we study human behavior and brains that cause this behavior (which had up until this point been arguably the most complex 'black boxes' in the known universe).

This situation highlights a fundamental clash in the two cultures of scientific model building: the traditional method of constructing models from the ground up by human experts versus the modern method of building from the data up by self-learning algorithms. This dichotomy has perhaps nowhere been as succinctly encapsulated as in Fred Jelinek's tongue-in-cheek remark (speaking in 1988, long before witnessing the advances in modern AI): 'Every time I fire a linguist, the performance of the speech recognizer goes up.'

Relying less on hyper-specialized domain expertise also promises more flexibility and scope for cooperation across fields: AI is fostering a new kind of scientific 'lingua franca,', enabling scientists from different fields to communicate in the shared language of latent variable models, autoencoders, or ML packages like `Pytorch`.

On the other hand, the dominance of large, opaque architectures in SOTA models complicates the extraction of knowledge from these models. From a scientific perspective, these models are often ill-suited to provide insights into their inner workings. In fields like neuroscience, psychology, or medicine, data is further much more difficult to come by than in areas like language or image processing, where vast amounts of data are readily available on the internet.

In scientific ML, where interpretation of inferred models is often of primary interest [280] and data is sparse, other approaches are therefore needed, and the question arises how models can be designed to optimally incorporate human-derived scientific knowledge as an inductive bias while retaining the flexibility and expressivity of modern ML approaches.

The field of dynamical systems reconstruction (DSR), and the methods developed during this thesis, can be seen as a bridge between traditional and AI-driven approaches. DSR increasingly relies on ML methods for the extraction of models from data, especially in contexts where dynamics are too complex to be captured by simple equations or where the underlying principles are not fully understood, such as in neuroscience or climate science. On the other hand, in DSR there is a strong emphasis on the interpretability of inferred models for further scientific insight [97], and an increasing emphasis on the integration of prior domain knowledge, such as in physics-informed ML [321].

The RNN models developed in this thesis were specifically designed with their mathematical tractability in sight (Sect. 3.1), and are trained based on theoretical insights into the nature of chaotic DS (Sect. 3.2). However, at the same time, they can be used in a range of different contexts, from complex chaotic weather systems (Sect. 4.2.1), experimental multimodal neuroscientific data (Sect. 4.3.3) or discrete behavioral data from psychological experiments (Sect. 5), with their analysis tailored to the specific research questions at hand.

2.2 DYNAMICAL SYSTEMS

Dynamical systems (DS) underpin many real-world systems of scientific and practical importance. Complex chaotic and multi-fractal DS are believed to govern market dynamics [263], as well as the rhythms of the brain, efficiently encoding information across brain regions while simultaneously solving numerous tasks [62]. Chaotic DS also form the basis of weather and climate models, from Edward Lorenz's model of atmospheric convection, which introduced the famous chaotic butterfly attractor and spurred the scientific investigation of chaos [129, 252], to models of climate patterns such as El Niño [400].

The concept of DS goes back to the 19th century, with Poincaré laying important groundwork [191, 315] for the formal study of DS. Following the definition in [139], a DS is defined by a combination of a state space $M \subseteq \mathbb{R}^n$, a time set $\mathcal{T} \subseteq \mathbb{R}$, and a law that governs its evolution over time. For a continuous-time system, this evolution is described by ordinary differential equations (ODEs):

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in M \subseteq \mathbb{R}^n, \tag{1}$$

where the vector field (VF) $f \in \mathfrak{X}^1(M)$ comprises functions with continuous first derivatives on the compact, metric, measurable state space M. This VF determines the evolution of initial conditions in time via the evolution operator $\Phi : \mathcal{T} \times M \to M$, which maps an initial condition $\boldsymbol{x}_0$ to states at time t, $\boldsymbol{x}_t = \Phi(t, \boldsymbol{x}_0)$ [220]. Even if such a solution exists in principle, most DS can not be solved analytically [308]. For systems where the evolution depends not only on time but also on spatial coordinates, partial differential equations (PDEs) are used:

$$\dot{\boldsymbol{u}} = g(\boldsymbol{u}, \nabla \boldsymbol{u}, \Delta \boldsymbol{u}, \dots), \quad \boldsymbol{u} \in M \subseteq \mathbb{R}^{n+1}, \tag{2}$$

where g represents a function involving spatial derivatives such as gradients (∇) and Laplacians (Δ).

While these formulations assume a continuous-time DS, empirically we usually access DS via discrete-time measurements, and thus for many empirical DS models, formulations as discrete-time maps are more natural [58]:

$$\boldsymbol{x}_t = \boldsymbol{F}(\boldsymbol{x}_{t-1}), \qquad \boldsymbol{x} \in M \subseteq \mathbb{R}^n. \tag{3}$$

The relationship between discrete and continuous time systems is rich and complex, so doing it justice would go beyond the scope of this introduction. While some discrete-time systems can be reformulated as continuous DS [282], this transformation is usually not unique. The reverse direction, sampling a discrete system from a continuous one, can for instance be achieved by discretely sampling the trajectory in time [58]. Assume we have datapoints x_k sampled from

the DS at intervals $k\Delta t$, such that $x_k = x(k\Delta t)$. Then, the discrete-time system is characterized by the mapping $F_{\Delta t}$, which is parameterized by the time step Δt. This induces a flow map $F_{\Delta t}$ for the discrete step Δt that maps the system to time $t_0 + \Delta t$, given by

$$\boldsymbol{F}_{\Delta t}(x(t_0)) = x(t_0) + \int_{t_0}^{t_0+\Delta t} f(x(\tau))d\tau. \tag{4}$$

An important tool for representing continuous DS as discrete maps is the Poincaré map [392]. This involves selecting a subset of the system's phase space, known as a Poincaré section, and observing the points at which a system's trajectory intersects this section. By mapping the points on this section onto each other, a continuous DS is effectively reduced to a discrete map that exists in a state space with one dimension less than the original system. Properties of the continuous DS, such as the stability of its orbits, can then be related to properties of the Poincaré map, such as the stability of its fixed points.

Chaos can manifest in discrete-time systems with two (or even one) dimensions, such as the logistic map. However, in continuous systems, a minimum of three dimensions is required to produce chaos due to the Poincaré-Bendixson theorem [74]. Discrete DS can also feature different kinds of bifurcations than their continuous counterparts, such as the Neimark-Sacker bifurcation [219] in maps, referring to the birth of a closed invariant curve (a periodic orbit) from a fixed point as a parameter of the DS is varied, similar to a Hopf bifurcation in continuous time systems.

CHAOS Chaos is ubiquitous in real world systems [91, 102, 108, 186, 264]. While there are several related formal definitions of chaos [43], they are unified by describing systems that are highly sensitive to initial conditions and feature some bounded, but not periodic, motion. Sensitivity to initial conditions is the most important practical condition for estimating the presence of chaos and implies that even small changes in a system's states, induced by process or measurement noise or numerical errors (see Fig. 1) can lead to large divergences between system states. The divergence of trajectories due to sensitivity to initial conditions can be formally described by the maximum Lyapunov exponent λ_{max}. Considering two trajectories $x(t)$ and $x(t) + \delta x(t)$ with initial difference $\delta x(0)$, the maximum Lyapunov exponent $\delta x(t)$ describes how the norm of the difference vector evolves:

$$\lim_{t\to\infty} \lim_{\|\delta x(0)\|\to 0} \frac{1}{t} \ln\left(\frac{\|\delta x(t)\|}{\|\delta x(0)\|}\right) = \lambda_{max} \tag{5}$$

A positive maximum Lyapunov exponent ($\lambda_{max} > 0$) signifies that the trajectories will diverge over time, indicating chaos. $\lambda_{max} = 0$ corresponds to limit cycles, while $\lambda_{max} < 0$ denotes equilibrium points. Exponential trajectory divergence in chaotic DS is closely related to challenges encountered when training DSR models on time series from chaotic DS, which will be discussed in depth in Sect. 3.2.1. Numerical methods for approximating Lyapunov exponents from DS models are described in Sect. 4.1.1.

The Lyapunov spectrum of a system can also contain multiple positive Lyapunov exponents (hyperchaos). These encode the amount of entropy production of the system, where its entropy production is proportional to the sum of its positive Lyapunov exponents [309], and the entropy of the distribution over its limit sets can be used to quantify the complexity of its dynamics [139].

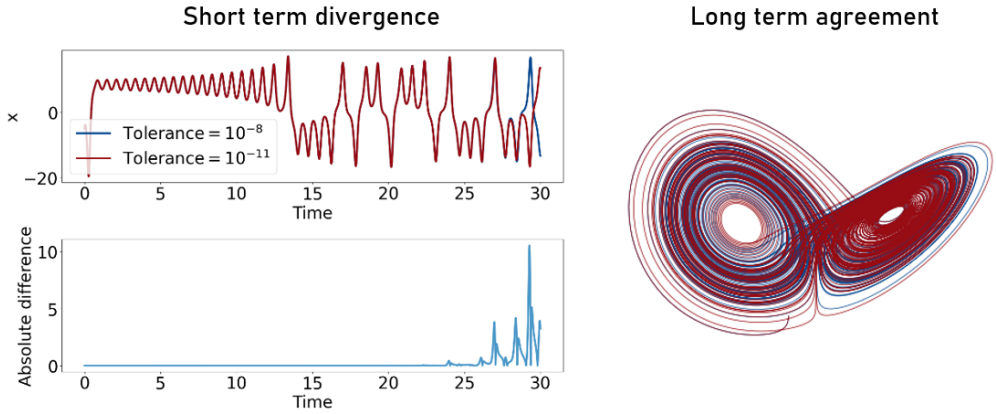

Figure 1: Illustration of chaos at the example of two simulated solutions of the chaotic Lorenz-63 system (Eq. 110) using the same numerical solver (Runge-Kutta method of order 5(4) [87] from `scipy.integrate`), with the sole exception being a small difference in absolute error tolerances (10^{-8} vs. 10^{-11}) of the numerical integrator. While trajectories diverge rapidly after a certain time horizon, the long-term limit sets still closely resemble each other.

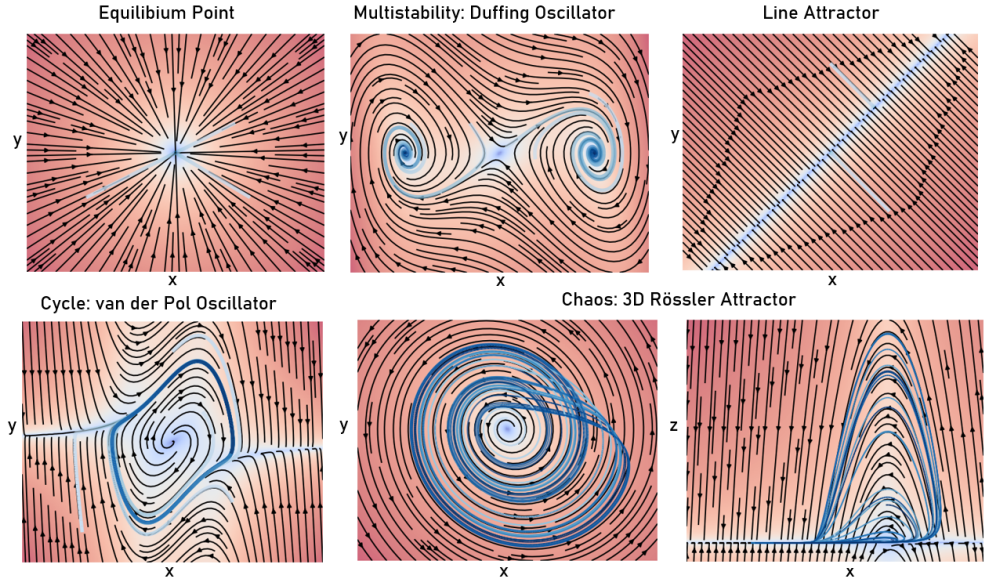

Figure 2: Flow fields and example trajectories (blue) for different 2D and 3D attractors.

ATTRACTORS AND LIMIT SETS Since the exact solutions of most DS can only be approximated numerically [308], and due to the presence of chaos in many real-world systems, global descriptions of DS are often more desirable than precise short-term predictions. These descriptions encompass the long-term set-theoretical and topological properties of the DS. Many of the performance measures for assessing the reconstruction quality of DS introduced in Sect. 4.1 are based on these considerations. One important concept is the ω-limit set of a point $\boldsymbol{x} \in M$. This set consists of the points reached when iterating the DS forward in time towards infinity:

$$\omega(\boldsymbol{x}, \Phi) = \bigcap_{s \in \mathbb{R}} \overline{\{\Phi(t, \boldsymbol{x}) : t > s\}} \tag{6}$$

Statistical errors based on (dis-)agreement of limit sets can be used to assess whether the reconstruction of a DS was successful (Fig. 3).

A related important concept in the context of a DS is that of an attractor. Following [139, 278, 308], an attractor is defined as a closed invariant set $A \subseteq M$ such that there exists an open and forward-invariant set $B(A) = \{\boldsymbol{x} \in M | \omega(\boldsymbol{x}, \Phi) \subseteq A\}$, called the basin of attraction, with $\omega(B(A)) = A$, and where A is minimal (i.e., there is no proper subset with that same property). Intuitively, the basin of attraction associates points in state space with an invariant set that fully captures its long-term behavior. In the simplest case, this can be a globally attractive equilibrium point but also allows for more complex objects, such as strange attractors. Fig. 2 illustrates several examples of attractors with different topologies. Together, the basins of attraction, combined with the topology of the attractors, can be viewed as encompassing the anatomy of a DS. One way of formalizing this anatomy with respect to its basins is by way of the Morse decomposition [139, 278]:

$$M = \sqcup_{e=1}^{n} B(A_e) \sqcup \tilde{M} \quad \text{such that} \quad \mu(\tilde{M}) = 0, \tag{7}$$

where the DS is composed of n disjoint basins of attraction, and μ is the Lebesgue measure, implying that this decomposition encompasses the entire state space (since the complement has measure zero). Many of the benchmark systems used in this thesis are globally attracting (where the basin of attraction spans the whole state space M). While DS can also feature multiple basins, as discussed in more detail in the context of multistability and generalization in Sect. 2.5.

2.3 DYNAMICAL SYSTEMS RECONSTRUCTION (DSR)

DSR is by now a rapidly growing field in scientific ML. In DSR, the goal is to learn a DS model from observations that constitutes a surrogate model of the data-generating DS. This entails that observations produced by the reconstructed model preserve important temporal, geometrical, and topological properties of the underlying DS. The general procedure is illustrated in Fig. 3. If such a surrogate model is successfully learned, it can be analyzed further for scientific insight [58, 97]. For instance, in neuroscience, computational mechanisms are believed to be implemented in terms of system dynamics [97], and inferring these dynamics can hence advance our understanding of the workings of the brain. In meteorology and climate science, one important goal is to predict the future state of a system, such as short and mid-term weather patterns [222] or long-term climate changes. In the context of model-predictive control [27], models can be used to analyze how a system is affected and controlled by external influences [110, 188]. A DS perspective is also becoming increasingly relevant in robotics [4, 89, 194, 432] for the control of complex movements in space. Finally, models can also be used for state estimation from limited or noisy measurements. This is exemplified by the Kalman filter's [184] contribution to Apollo 11 landing on the moon, or the important role state space models play in neuroscience [299, 385].

OBSERVATION FUNCTIONS AND TEMPORAL DELAY EMBEDDINGS

DS are usually empirically accessed via time series measurements $\boldsymbol{X}$, where $(\boldsymbol{x}_t)_{t=1\ldots T}$, $\boldsymbol{x}_t \in \mathbb{R}^N$. In DSR we often assume that measurements at time t are in some way related to an unknown (latent) unobserved dynamical process $\boldsymbol{z}_t$ via some measurement function $\boldsymbol{x}_t = h(\boldsymbol{z}_t)$ depending on the current time point or a window

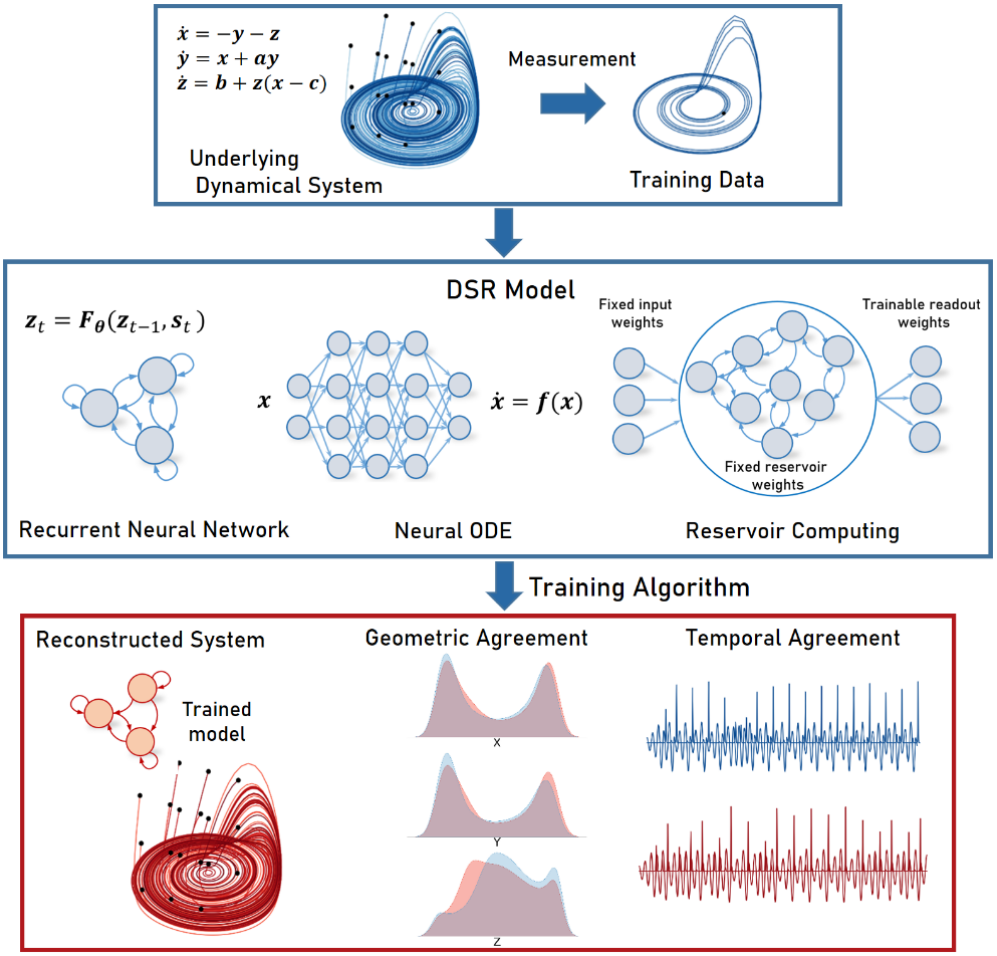

Figure 3: Overview over dynamical systems reconstruction.

of time points. This introduces two difficulties: first, the set of time points during which observations are measured can be relatively sparse in time, significantly undersampling the DS, or irregularly sampled. Second, the measurement function can only capture limited aspects of the latent dynamical process. For instance, in neuroscience, the primary object of interest, the brain, is hidden behind the skull, evolved precisely to keep intrusions like curious neuroscientists out. Imaging brain activity directly requires invasive techniques only carried out on animals, or, in rare cases, on patients undergoing invasive neurosurgery. Noninvasive techniques like functional magnetic resonance imaging (fMRI), on the other hand, require the application of strong magnetic fields in expensive machines and only resolve neural activity at relatively low temporal and spatial resolutions. While recording techniques are undergoing significant progress through the increasing availability of large-scale, high-density multi-electrode arrays and the rise of optogenetics, ushering in what Liam Paninski calls a 'golden age of statistical neuroscience' [192], even those techniques only image small parts of the intricate DS that is the brain as a whole.

One avenue of hope that important properties of underlying DS can still be reconstructed even from partial observations are the temporal delay-embedding (TDE) theorems [297, 349, 388]. A TDE is defined as:

$$\mathbf{x}_t = (\mathbf{x}_t, \mathbf{x}_{t-\tau}, \mathbf{x}_{t-2\tau}, \ldots, \mathbf{x}_{t-(m-1)\tau})$$

here τ represents the delay time and m the embedding dimension. To construct a TDE, one must first determine the appropriate values for τ and m. The parameter τ determines the temporal separation between the components of the embedding, while m determines the dimensionality of the reconstructed phase space. Optimal values are important for properly unfolding the attractor in the reconstructed state space, as illustrated in Fig. 4. Standard techniques for determining optimal values of these parameters include methods such as the first minimum of the autocorrelation or mutual information function [2, 187] for τ, and false nearest neighbors methods for m. More sophisticated techniques, such as the PECUZAL algorithm [215] exist for determining embeddings with different lags τ_m for each embedding dimension, and were used for the reconstructions of electrocardiogram data in Sect. 4.2.2. One important implication of the delay-embedding theorems is that for sufficiently large embedding dimensions, the reconstructed state space will reproduce all topological properties of the underlying system.

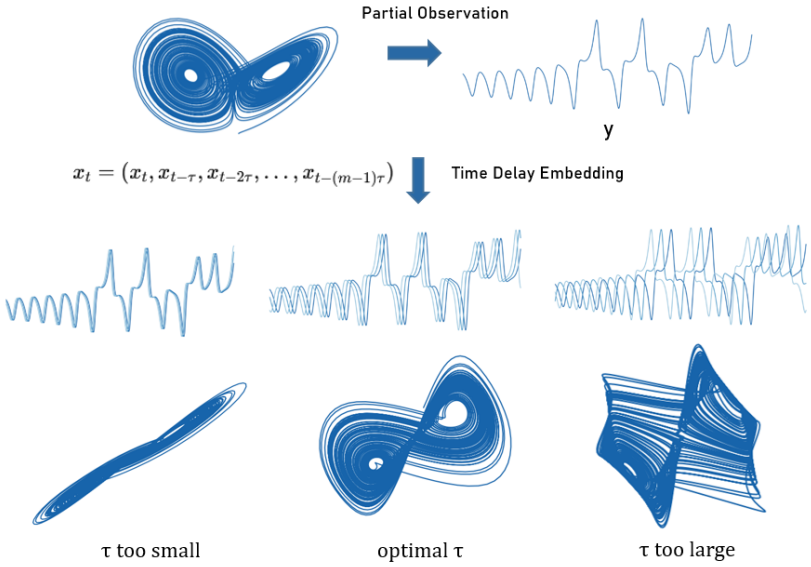

Figure 4: Illustration of temporal delay embeddings.

Another complication for DSR from empirical time series arises when considering that measurements of systems are often not continuous. For instance, survey data in psychology is quantified using Likert scales given by discrete ordinal levels [241]. Neuronal activity is often captured as spike counts, climate data comprises discrete events, and language is represented through distinct tokens [342]. While in most cases, we can assume that an underlying continuous process is causing these observations, it is often not clear how the measurement function $\boldsymbol{x} = h(\boldsymbol{z})$ distorts and coarse-grains the underlying process, and thus not clear how much of the underlying system can still be reconstructed. While these questions are of long-standing interest in the literature [347, 348], with [347] showing DSR is in principle feasible from interspike intervals in a neuroscientific context, many theoretical and practical questions surrounding the influence of measurement functions on DSR have not been resolved yet. The results in Sect. 4.3.2 present DSR from symbolic or otherwise highly coarse-grained representations of chaotic DS. While providing evidence that reconstructions are often still feasible even under these challenging circumstances,

clear theoretical results underpinning under which conditions these reconstructions remain feasible are still lacking.

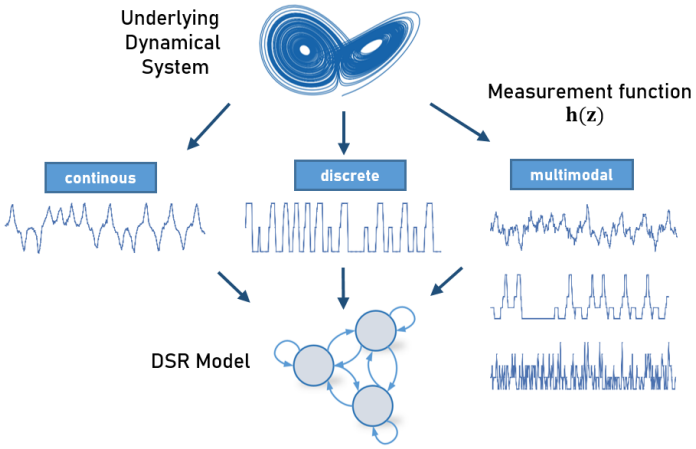

Figure 5: Illustration of measurement functions in DSR.

The ideas behind Koopman theory [57, 208] and its operators are closely related to the discussion of measurement functions and the challenge of capturing the dynamics of latent DS through observations. Koopman theory can be seen as an 'operator-theoretic perspective' on DS theory [58]. Given a DS represented by the evolution function $z_t = F(z_{t-1})$ as before, the Koopman operator U acts on an infinite-dimensional Hilbert space[1] of observation functions $h : \mathbb{R}^N \to \mathbb{R}$. For any observation function h, the action is defined as:

$$(U_h)(z) = h(F(z)). \tag{8}$$

This implies that the Koopman operator U transforms the observation function h based on the system's evolution function f via a linear operator operating in the infinite-dimensional space of observation functions. It hence allows for the analysis of nonlinear DS using linear techniques. While the infinite-dimensional nature of the Koopman operator also introduces practical challenges, working in linear representations of nonlinear DS makes them much easier to analyze, predict, and control [57]. Accordingly, many techniques, including many ML-driven approaches, have been developed in recent years to approximate the action of the Koopman operator in finite dimensions in the context of DSR [18, 56, 119, 257, 285, 296, 415].

The considerations outlined in this paragraph more generally point to the dual challenge inherent in data-driven DSR (and in many other scientific disciplines): algorithms must jointly discover coordinates/embeddings that allow for the representation of interpretable and generalizing models in this coordinate system [218]. The MTF framework introduced in Sect. 3.2.5 tackles this challenge by learning DSR models within a flexible encoder-decoder structure, inferring an appropriate coordinate representation of the underlying DS even for situations in which the measurement process itself is highly complex, such

1 Koopman theory also bears some resemblance to quantum mechanics (QM): in QM, quantum states live in infinite-dimensional Hilbert spaces, and observables are represented as self-adjoint operators acting on these spaces. The evolution of a quantum state is determined by the Schrödinger equation, which, similar to the Koopman approach, is linear, so it is perhaps no surprise that Koopman and von Neumann developed this theory in 1932 during the golden age of QM [208].

as when training on ordinal samplings or symbolic representations of the underlying DS (see e.g. Sect. 4.3.2, Fig. 29), or even discovering an appropriate TDE when training on partial observations (Fig. 27).

DS MODELS AS PROBABILISTIC GENERATIVE MODELS Generative models are statistical models that model joint probability distributions over variables, often to produce new data following an observed distribution $p(\boldsymbol{X})$. Deep generative models, combining generative approaches with deep learning architectures have taken the world by storm. These include GANs [132], variational autoencoders [202, 330], diffusion models [371] or transformer-based LLMs [406], which have found widespread use in practical applications [292, 323]. Following the assumption that measurements of a dynamical system $\boldsymbol{X} = \boldsymbol{x}_{0:T}$ depend on an underlying unobserved latent process $\boldsymbol{Z} = \boldsymbol{z}_{0:T}$, a generative model of an observed time series $p_\theta(\boldsymbol{x}_{0:T})$ can be written as [127]:

$$p_\theta(\boldsymbol{x}_{0:T}) = \int p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{z}_{0:T})p(\boldsymbol{z}_{0:T})d\boldsymbol{z}_{0:T}. \tag{9}$$

where the $\boldsymbol{X}$ depend on some underlying latent process via the distribution $p_\theta(\boldsymbol{X}|\boldsymbol{Z})$, which can be viewed as the probabilistic formulation of the measurement function $\boldsymbol{x} = h(\boldsymbol{z})$. For DS, we usually assume that the prior is autoregressive, and only depends on the τ past observations. More explicitly, assuming that the prior is given by a discrete-time DS (Eq. 3), it becomes a first-order hidden Markov model, in which each future state $\boldsymbol{z}_{t+1}$ is conditionally independent of the past given the present $\boldsymbol{z}_t$:

$$p_\theta(\boldsymbol{z}_{0:T}) = p(\boldsymbol{z}_0)\prod_{t=1}^{T} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}). \tag{10}$$

It is often assumed that observations at any time t conditionally only depend on the current latent state $\boldsymbol{z}_t$, which further simplifies the likelihood function:

$$p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{z}_{0:T}) = \prod_{t=0}^{T} p_\theta(\boldsymbol{x}_t|\boldsymbol{z}_t)d\boldsymbol{z}_{0:T} \tag{11}$$

However, more complex relationships between dynamics and observations, such as those given by hemodynamic response function in fMRI, are also possible [212].

This class of models, consisting of a state equation and an observation equation, is usually referred to as state space models (SSMs, [95]). If dynamics are linear and observations and latent states are Gaussian, the Kalman filter [184] represents the optimal solution for estimating the state of a linear system with Gaussian noise. If these assumptions are not met, e.g. for nonlinear DS, more sophisticated approaches such as Extended Kalman Filters [332], Unscented Kalman Filters [413], Expectation Maximization [95, 212] or Variational Inference (VI, [202]) can be used. The RNN models introduced in Sect. 3.1 can be naturally formulated as SSMs by assuming the nonlinear latent variable model is given by an RNN (Eq. 12 and coupled to different observation models (Sec. 3.2.7), and trained using e.g. VI (Sect. 3.2.2) and MTF (Sect. 3.2.5).

2.4 MACHINE LEARNING TECHNIQUES FOR DSR

Given the advances in ML techniques to learn models from data, the field of DSR has likewise seen the development of many novel approaches in the past decade. There are several angles through which a classification of different DSR methods can be approached. An important line of distinction is for instance between continuous-time models that approximate the vector field of the DS directly [59, 71, 148, 368], while discrete-time models, such as those based on recurrent neural networks (RNNs) [95, 304, 355, 444] approximate the observed time series as a discrete-time map. Vector field-based approaches often require first numerically estimating derivatives from the data before inference, which can be noise-prone and introduce additional numerical instability [25, 322]. On the other hand, they more naturally accommodate irregularly sampled time series [336, 438], which can often occur e.g. in experimental settings. Some approaches combine the advantages of both methods, e.g. by allowing the transformation of discrete-time formulations into continuous time models [282], or by estimating the vector field while training directly on the observations, such as in Neural ODEs [70, 93, 336].

The 'classical' scientific paradigm, reaching back to the days of Kepler and Newton, relies on symbolic regression of model coefficients. Symbolic approaches have seen increasing popularity as a tool for data-driven model discovery [221, 261]. Since they are formulated in the same language as scientific models, one key advantage is their interpretability [148], making them useful in scientific applications or when used e.g. in high-stakes decisions [339]. In DSR, the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm and its many variants [59, 76, 182, 183, 251, 274] represent the most widely used regression-based approach. These methods perform (sparse) regression on predefined function libraries on estimates of the vector field directly (for more details, see Sect. A.2.1). However, symbolic approaches have limitations, including their difficulty in capturing complex and noisy empirical data ([54, 152], see also Sect. 4.2.2). They further often require some prior knowledge of the system's underlying structure, limiting their applicability in discovering novel phenomena or in fields where the system dynamics are not well understood. Recent approaches aim to overcome these limitations by integrating symbolic regression with more flexible ML techniques, such as neural networks (NNs, [198]) or transformers [449].

On the other hand, many recent DSR approaches rely on the principle of universal approximators, an important concept in ML that states that sufficiently large NNs with at least one hidden layer can approximate any function to arbitrary precision [158]. Importantly, these results have been extended to RNNs, establishing that sufficiently large RNNs can approximate any underlying DS [116, 146, 200]. Several different classes of DSR methods based on universal approximators exist, featuring different model formulations and training paradigms. Following the grouping in [97], there are broadly speaking three such classes used in DSR: those based on RNN models that are often combined with special training techniques, e.g. backpropagation through time (BPTT, [54, 408, 409]). RNN models also often feature specific model architectures or regularizations [66, 106, 181, 193, 341, 355] or are combined with specific training techniques [276] to remidy exploding-and vanishing gradient problems [33, 54, 152, 276]. Another class of models is given by reservoir computers (RC, [304, 312, 313]), featuring a large randomly initialized reservoir while training only

a read-out layer via linear regression to the reservoir dynamics. Lastly, there are neural ODEs [11, 70, 189, 204], which approximate vector fields directly via NNs, and which are usually trained using the adjoint method [70]. Many of these techniques have been evaluated as comparison methods for this thesis, and are described in more detail in Appx. Sect. A.2.

This grouping is, however, not exhaustive, as DSR algorithms vary across several criteria that may significantly influence their applicability to specific problems. Many DSR algorithms can, for instance, be formulated both as deterministic or probabilistic algorithms and often both variants of similar algorithms exist (see e.g. SINDy [59] vs. HyperSINDy [171], Neural ODEs [70] vs. Neural SDEs [399] etc.). Often models also try to separate deterministic and stochastic components, such as in Langevin-type SDEs [171, 228] which differentiate between deterministic drift and stochastic diffusion terms. This thesis likewise will introduce and evaluate both deterministic (Sect. 3.2.3 & 3.2.4) and probabilistic (Sect. 3.2.2 & 3.2.5) algorithms for DSR. In many applications such as in neuroscience, quantum physics, or finance, probabilistic approaches that explicitly model observation and dynamic noise are often more suited to represent the uncertainty inherent in real-world systems, and naturally provide uncertainty estimates, which are often desirable e.g. in a clinical context ([232], see also Sect. 5). Optimizing probabilistic models may often present additional challenges during training, motivating the search for optimal trade-offs [121].

TIME SERIES FORECASTING While often not the primary goal, DSR models can be naturally used to forecast time series, and hence approaches in time series forecasting (TSF) often share commonalities with DSR algorithms. While a DSR model aims to learn the latent DS $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$ underlying an observed time series and the relationship between the latent process and the observations $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$, the goal of TSF is to learn a probability distribution over a state based on its past, $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}...\boldsymbol{x}_0)$. If a DSR model fully captures the DS underlying the TS, it naturally serves as an optimal forecasting model [126]. However, since a good DS model is challenging to learn, particularly from empirical data, TSF algorithms often approximate the distribution over future time steps directly from the history of past time steps. A sequential latent variable model is not strictly necessary for this, and TSF can be approached by training sequence-to-sequence mappings without explicitly adopting the inductive bias of a sequential approach, such as the factorization of the probabilistic model according to Eq. 10.

Since TSF is a prominent problem with many practical applications, from weather forecasting to stock market predictions, it has a long history that reaches back long before the advent of ML methods. Classical methods, such as Autoregressive Integrated Moving Average (ARIMA) models [49], predict the future value of a system using a linear combination of its past values and errors. With RNNs being the most popular choice for sequential data for many years, many RNN-based architectures have been employed in TSF [324, 345]. However, they usually require training with BPTT, which leads to exploding-and vanishing-gradient problems (EVGP) omnipresent in gradient-based training on sequential data [33] (related problems are discussed in more depth in Sect. 3.2.1). When not relying on a sequential model, these problems can be avoided. The success of transformers has largely replaced sequential models like RNNs in other domains such as language [250, 406, 448], and has led to many publi-

cations using transformer models for TSF [45, 346, 425, 446]. Unlike sequential models, transformer models are easier to parallelize on GPUs though they bear a high computational load, with their training cost scaling naively with $\mathcal{O}(T^2)$ given the sequence length they are trained on. Given their widespread use in commercial applications, the search for more computationally scalable sequence models remains an active area of research. One novel emergent class of models is structured SSMs [135, 136, 370]. These retain the inductive bias of underlying continuous-time dynamics (or discrete approximations thereof) to model sequences, similar as in Eq. 9, but usually assume linear dynamics to allow for more efficient parallelizable optimization as in transformers, e.g. by combining linear layers with fast Fourier transforms or by using structured sparse matrices. In combination, they reduce the computational complexity of transformers from $\mathcal{O}(T^2)$ to $\mathcal{O}(T \log T)$ or even $\mathcal{O}(T)$. Similarly, approaches based on Convolutional NNs (CNNs) [213, 233], which can be optimized on long sequences in parallel, are widely employed in TSF and TS classification [168]. While some publications claim that ML methods have become SOTA for TSF [34], others have put the validity of using transformers for TSF into question, showing that they are often outperformed by simpler models [440], and only excel when large amounts of clean data are available. Popular TSF competitions, such as the M5, have routinely seen a prevalence of methods not based on deep learning techniques, such as those based on trees (e.g. random forest or gradient boosted trees), among the best-performing models [175].

A different but related subfield that aims to model complex dynamics with ML methods involves approximating reduced order models [10]. Reduced order models are particularly relevant when an existing DS of a complex system, such as a fluid flow or the Earth's weather, has to be simulated on supercomputers, incurring prohibitively high computational costs. Fourier Neural Operators (FNOs) have gained popularity in this context recently [235, 254]. Similar to other sequence-to-sequence models, these don't directly approximate the system dynamics represented e.g. by a set of PDEs, but instead learn to numerically approximate their solutions given parameterizations and initial conditions. The FNOs are then trained on numerically simulated solutions, leading to relatively high training costs, but end up with efficient representations that incur much lower inference costs. FNOs have also been combined with physics knowledge [237] and extended to other settings, such as deformed grids [234], and applied to model chaotic DS [236]. As in structured SSMs like S4 [136], moving to Fourier space avoids the costly scaling of training sequential models, since in this representation computations run in parallel and often numerically efficient approximations exist.

A variety of leading generative modeling approaches, including score-based methods [373], continuous normalizing flows [70], continuous diffusion models [86], and flow matching [247], integrate concepts from DS theory. For instance, diffusion models were inspired by diffusion processes in non-equilibrium thermodynamics [371]. These methods aim to learn continuous deformations of simple probability distributions into complex target distributions, typically the data likelihood $p(\boldsymbol{x})$, such as the probability over all images in a training set. These deformations are often easier to learn with gradient-based methods than approximating the data likelihood directly. The evolution of these distributions can be viewed as trajectories in the space of probability densities, and their evolution is governed by (Neural) ODEs and SDEs. Since for valid probabilistic

modeling, probability mass has to be conserved, physical constraints inspired by fluid dynamics like the continuity equation are often integrated [247].

MULTIMODAL DSR Multimodal data integration is a popular topic in many areas of AI and ML [5, 15, 23, 39, 239, 246, 292, 365, 379, 387, 417, 431], particularly since the advent of models embedding vision and language in joint latent spaces [13, 320]. Multimodal generative models combine multiple observed modalities into a common latent representation, an ability not unlike the brain fusing different sensory inputs (vision, hearing, touch, etc.) into a combined world model [114]. Multimodal integration can be used for cross-modal prediction or output generation (such as generating textual descriptions of scenes). It can also be used to complement noisy or missing information in one channel through observations from other modalities [319]. Humans routinely do this when combining auditory and visual signals in noisy environments, e.g. when combining visual cues from lip movement to understand each other in noisy environments. Discovering joint multimodal representations can further reveal interesting links among observed modalities [238], e.g. when combining different data streams such as genomics and imaging to improve diagnostics in clinical settings [246]. Variational autoencoders (VAE) [202, 330], introduced formally in Sect. 3.2.2, are one popular variant of generative models which naturally lend themselves to multimodal extensions [23, 387, 426] and applications to time series data [20, 26, 128]. VAEs have been applied to multimodal time series data for TSF [15, 39, 85, 319, 365, 387].

Despite the widespread use of multimodality in other areas of ML, a generally effective framework for multimodal data integration in the context of DSR is still lacking. Kramer et al. [214] proposed the first work of this sort, training DSR models with two algorithms based on sequential VAE (Sect. 3.2.2) and an Expectation Maximization (EM) algorithm [95, 212]. However, both algorithms come with significant downsides: while the SVAE performs poorly on experimental data, as observed in [214] and [54], the EM algorithm incurs high training costs and requires adjustments to the training objective for different combinations of multimodal data. These shortcomings motivated the development of the multimodal teacher forcing (MTF) approach introduced in Sect. 3.2.5 as a general and effective framework for DSR from multimodal observations.

2.5 GENERALISATION IN DSR

The task of generalizing from training data to unseen test data lies at the heart of much of ML [414, 441]. However, in the field of DSR, this question has only been explored to a limited extent. This observation might be related to the fact that benchmark validations in DSR are still mostly carried out on simple synthetic benchmark systems. To this end I carried out a survey of 59 papers in the field of DSR for [139], based on a wide range of methods and applications, classifying them based on which benchmark datasets were employed. Benchmark datasets were classified according to the following three categories:

- Simple linear or nonlinear systems like the Fitz-Hugh-Nagumo and Lotka-Volterra systems, or coupled or damped harmonic oscillators/pendulums like the van-der-Pol oscillator

- simple monostable chaotic attractors, predominantly the Lorenz-63, Rössler, or forced Duffing attractors
- nonlinear PDEs, inspired e.g. by fluid flows or convection (Burgers equation, Navier Stokes equation, Lorenz-96, Kuramoto–Sivashinsky equation, etc.)

Category	Counts
Linear Models/Oscillators	24
Chaotic 3D Models	29
Fluid Dynamics/PDEs	13
Experimental Data	6
Multistable	3

Table 1: Classification of benchmark systems in the DS literature.

Table 1 summarizes these results. Only a small part of DSR considers experimental datasets as benchmark systems, half of them being in publications from our group, and almost no publications consider multistable systems. SOTA DSR algorithms by now often show good performance on monostable benchmarks like the Lorenz-63 or Lorenz-96 systems (see also Table 7), generalizing to nearby initial condition and reproducing the long-term temporal and geometric structure of the DS. However, for most of these systems, given their monostable nature, generalization does not extend to unobserved or underexplored regions of state space, especially if these regions feature different dynamics than those present in the observed data. Multistability, where multiple attractors coexist within the same system, is a common feature even in simple, low-dimensional nonlinear DS, and is believed to underlie many real-world systems, from neuroscience [99, 170, 197, 353], chemistry [288], and biology [92] to financial markets [65] and climate science [437]. One important insight in [139] is that multistability can be intimately tied to out-of-domain generalization (OODG), where models are required to generalize across basin boundaries. On the other hand, generalization for monostable systems can often be viewed as a form of in-distribution generalization (Fig. 6). Another challenge in DSR from experimental time series is the presence of external inputs to the system [385], or the presence of non-stationarities, such as slow changes in the earth's climate, or shifts in neurotransmitter concentrations in the brain [97] that can further complicate the discovery of generalizing solutions.

Generalization in DSR has only been indirectly studied in the literature. Of particular interest for instance in climate science is the anticipation of tipping points [61, 120, 302] and prediction of post-tipping point dynamics. A related task is the forecasting of extreme events, here denoting phenomena that are underrepresented in the training data but can have high practical relevance [109, 138, 318]. Other approaches model multiple environments with distinct parameter settings and jointly infer the environment's dynamics and assess to which environment an observed trajectory belongs [35, 436]. In recent years, there is also an increasing interest in physics-informed ML (PI-ML, [269, 321, 382]), which has been more widely applied in DSR. PI-ML integrates physical laws or domain knowledge into ML algorithms, enhancing the model's ability to generalize and perform well in situations where training data is sparse. It can also

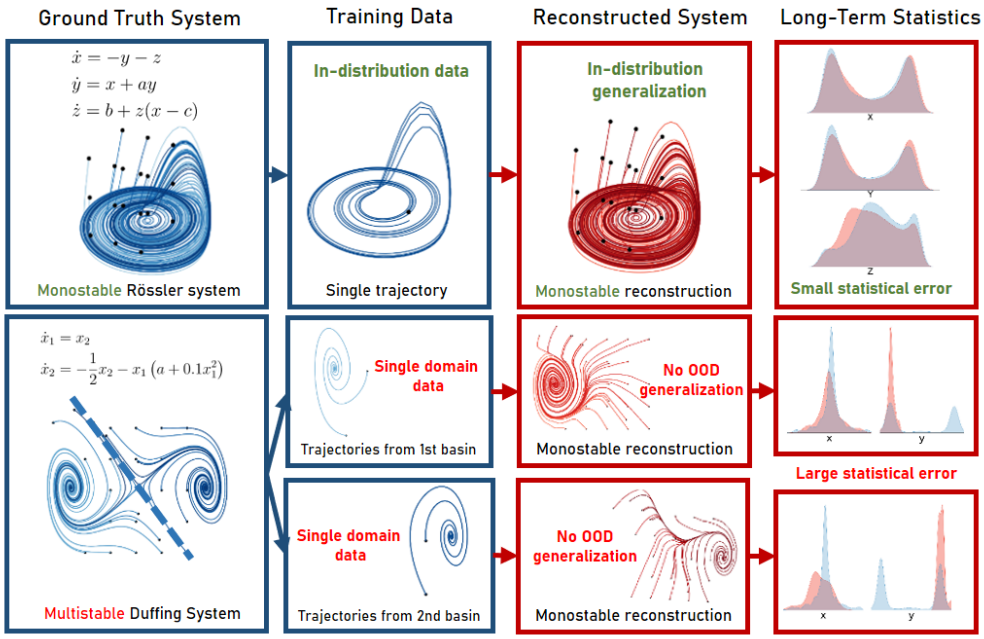

Figure 6: Generalization and multistability in DSR.

help build physical constraints into DS [185], e.g. when modeling power grids where unphysical predictions and errors can prove very costly [279]. However, to what extent DSR algorithms generalize to unobserved regions of state space remains an open challenge. In [139] we aim to address this by providing a first formal and empirical treatment of generalization in DSR (see also Sect. 4.1.3), investigating under which conditions common DSR generalize (or fail to) to unobserved data, particularly to data from unobserved basins in multistable DS.

2.6 HIERARCHISATION AND TRANSFER LEARNING IN DSR

Recent foundation models such as LLMs are trained on an abundance of data [292]. In contrast, many experimental or empirical settings encountered in the sciences, typically deal with much sparser data [211]. Data collection can be expensive and challenging, relying on expensive measurement devices or on self-reports that require active compliance from participants [419]. This scarcity of data often makes traditional learning algorithms difficult to apply, and more prone to overfitting. Conversely, in many scientific settings, data is collected from multiple 'subjects' $\boldsymbol{X}^N$, with 'subjects' denoting different instances of a process expected to display both shared structure and significant inter-individual variation. This can include measurements from several similar physical systems, a group of patients in a medical study [165, 404], or repeated measurements from the same subject across consecutive trials [439].

HIERARCHICAL MODELING Training individual models for each subject fails to 'transfer' group-level information across subjects [298], and the independence of model training complicates the quantification of inter-subject variability [439]. Hierarchical models tackle this challenge by integrating data across multiple levels of abstraction, allowing for the sharing of statistical strength

among subjects while maintaining room for subject-specific differences. One common approach is to extract a set of latent variables from models that quantify subject-specific differences [260], and that can be used to extract class labels for mental illnesses (e.g., healthy vs. schizophrenic, [211] or other interpretable structures [395]). Bayesian hierarchical models have been used as a principled framework for encoding shared group- and subject-level structure, and are usually trained using common statistical optimization techniques such as Monte Carlo techniques [38] or VI [202, 330]. The hierarchical structure of the data can be directly incorporated into the probabilistic structure of the Bayesian model, e.g. by drawing individual-level parameters from shared group-level distributions. Bayesian hierarchical models usually define priors for each parameter and therefore naturally incorporate prior knowledge or assumptions about the parameters into the model [118] (see also Sect. 3.2.2).

APPLICATIONS TO TIME SERIES AND DS Since time series are prevalent in many scientific and clinical settings, several hierarchical approaches have been adapted for time series data. [22, 447] introduce a dynamic hierarchical model that models individual time series through linear models while incorporating explicit dependencies to an average group-level model. Different versions of hierarchical models have been used for forecasting tasks [265, 266, 344, 430], or for extracting interpretable summary statistics to encode inter-individual differences between time series [6, 430]. Hierarchical models are conceptually related to the principle of transfer learning in ML [298], with ideas from transfer learning being increasingly applied to time series and DS models. Examples include one-shot learning of linear differential equations using physics-informed ML [84], using RCs [137] to learn two closely related DS, or transferring learned dynamics across different robots [374]. In a DS context, the tasks of learning new dynamics from small amounts of data by transferring knowledge from different dynamical regimes of the same system [167] and generalizing to new unobserved dynamical regimes [203] are conceptually closely related to the hierarchization approach that will be introduced in Sect. 3.3.

Inspired by the success of LLMs, the transfer learning paradigm is also encapsulated in the quest for foundation models that can be flexibly adapted to a range of specific tasks through one-shot/few-shot learning [448] and fine-tuning. Several attempts have been made to train foundation models for time series recently [277, 434], with the leading tech companies like Amazon [14], Google [81] and Meta [325] all releasing such models in the last few months. Since hierarchical approaches serve as a natural mechanism for transfer learning, they could also provide a mechanism to implement such foundation models tailored to DSR.

2.7 APPLICATIONS OF DSR

While the methods developed in this thesis are general, many of the developments were motivated by applications to real-world data. Within the context of the Theoretical Neuroscience and Computational Psychiatry group at the Central Institute of Mental Health Mannheim, our focus was on applications in neuroscience and psychiatry. Particularly, important parts of this thesis, such as the MTF framework (Sect. 3.2.5) and hierarchization framework (Sect. 3.3),

were specifically developed for the IMMERSE (Implementing Mobile MEntal health Recording Strategy for Europe) consortium, which aims to integrate data-driven approaches into psychiatry. As two motivating examples, I will therefore discuss applications of DSR in the context of neuroscience and psychiatry. To illustrate the wider relevance of DSR in other fields, I will also discuss applications in the context of quantum (many-body) physics.

DSR IN QUANTUM PHYSICS DS theory is intimately intertwined with many theories in physics, and ML approaches have been used to model DS in many disciplines. These applications range from approximating the evolution of quantum systems [161] and predicting the motions of galaxies in astronomy [207], to simulating fluid flows in hydrodynamics [206] and forecasting large-scale climate models [305].

For example, DS theory plays a critical role in describing the temporal evolution of quantum systems [156]. As noted in the context of Koopman theory, contrary to classical many-body dynamics, QM is linear, so there is no intrinsic chaos. However, because the Hilbert space dimension of many-body systems is exponentially large, QM systems are still hard to solve [179]. One goal of quantum many-body physics is to identify effective lower-dimensional descriptions, which are often non-linear [140], such as the Gross-Pitaevskii-equation describing Bose-Einstein condensates [134]. Even though QM does not feature chaos in the usual sense due to its linearity, it has recently been pointed out that chaos-like signatures can be seen in specific observables [8], a phenomenon also called the 'quantum butterfly effect'. For instance, [196] invoke a strong connection to classical chaos by finding positive Lyapunov exponents in commutators of quantum systems, which have also been used to study many-body quantum chaos near phase transitions [358].

While quantum computers are increasingly leveraged for the simulation of quantum systems [369], often effective classical thermodynamical descriptions called 'classical shadows' [161] of these systems have been shown to emerge [179, 358]. The development of these models, however, is often laborious and challenging, so data-driven DSR techniques provide a natural way to learn these descriptions directly from data [150, 161].

On the other hand, quantum computers are increasingly combined with ML algorithms to leverage the strengths of both approaches [32, 40]. Tasks that are hard to optimize on classical computers are sometimes more naturally suited to quantum computers [226], such as sampling from complex probability distributions, as common in modern ML. For instance, quantum versions of Markov Chain Monte Carlo methods [226, 255, 357] and stochastic gradient optimization [124] have been proposed, which integrate sampling from a quantum computer into the algorithms. Closely related to the methods discussed in this thesis, similar hybrid approaches have also been employed in quantum reservoir computers [122, 429] (where a quantum computer provides the dynamical reservoir) for modeling classical chaotic DS [429] and quantum systems [122]. More generally, quantum computers have been used to simulate complex classical DS with exponentially large dimensions [123], since computational tasks scaling exponentially on classical computers often scale in polynomial time in a quantum setting [367]. However, it remains an open problem how to extract the exponentially large information about the DS from the quantum computers, for which ML approaches could again be helpful.

DSR IN NEUROSCIENCE DS approaches are of long-standing importance in computational neuroscience, with computational processes in the brain believed to be implemented in terms of neural dynamics [97]. A famous early example is given by Hopfield networks [157], a type of NN model that mimics associative memory by learning to store and 'remember' patterns seen during training by implementing the memory of the patterns via fixed points of its dynamics. While the underlying dynamics of original Hopfield networks are quite simple, modern versions of Hopfield networks have been directly linked to neurobiologically plausible implementations of working memory [217].

DSR approaches offer the possibility to extract computational models from data directly, which can be leveraged on different levels to gain insights into underlying computational mechanisms. For example, a DSR model inferred on the individual neuron level can be used to simulate stimulus-response curves of the model to external currents. Interpretability of inferred models can further be enhanced by including substructure into DSR models, where e.g. a subsection of the latent states maps onto excitatory units and another on inhibitory units of a population [97], observation models are structured to encode different brain regions with different subsets of the latent states [97, 210], or topological structure is introduced into the DSR model [149]. Extracted DSR models can be used for the classification and diagnosis of neurodegenerative disorders, such as lateral sclerosis [395]. Another line of research integrates neuroscientific recordings with behavioral data into joint embedding spaces [356], which can help elucidate the computational mechanism underlying behavior (see also Sect. 4.3.3).

DSR IN PSYCHIATRY In line with the trends discussed in Sect. 2.1, research in psychiatry is moving increasingly beyond verifying specific psychiatric hypotheses, but instead aims to leverage flexible data-driven models that are directly incorporated into clinical settings [69], e.g. enabling early intervention before the onset of acute mental health episodes, and recommending tailored treatment strategies and interventions [153]. This includes integrating data from biomarkers such as genomics [12], or using digital measurement tools [159], such as smartphones or wearable devices, in the day-to-day life of subjects, moving from the traditional reliance on subjective assessments by psychiatrists towards the continuous monitoring of patients. Several studies integrate the collection of ecological momentary assessment (EMA) data, which consists of regular surveys reflecting the moment-to-moment well-being and symptoms of subjects, into psychiatric care [396]. This now often goes along with leveraging passively recorded sensor data e.g. by smartphones [230, 311, 333, 361, 393] for the extraction of features for the prediction and diagnosis of psychiatric symptoms. For instance, Place et al. [311] correlate features derived from data such as the sum of outgoing calls and absolute distance traveled with the severity of symptoms of mental health issues like depression or PTSD. Some recent studies try to combine EMA data with passive sensor data to forecast psychiatric symptoms with greater accuracy and in real-time [172, 173], such as predicting daily fluctuations of depressive symptoms assessed from EMA ratings together with GPS and weather data [172]. Some of these studies also leverage ML methods such as RNNs [210] and other deep learning techniques [69, 211] known for their potential to automatically extract complex patterns from many different types of data simultaneously, and also directly incorporate DS models [153, 210].

DS models are used based on the observation that psychiatric symptoms often follow a complex and individual temporal evolution [117]. Connecting both a neuroscientific and psychiatric perspective, mechanisms underlying mental illness can also be understood as changes in neural network dynamics [96]. For instance, [335] connect obsessive-compulsive disorder to overly stable neural attractors, while [334] propose that similarly deep attractors in the non-reward system in the lateral orbitofrontal cortex underlie depression.

Part II

METHODS

3

NETWORK MODELS AND TRAINING ALGORITHMS

This section recapitulates the main methodological contributions of Brenner et al. [54], Hess et al. [152] and Brenner et al. [53]. The developments in [152] build on the experimental results in [54], while [53] extends them to multimodal and non-Gaussian data. Section 3.1 introduces two novel RNN architectures, the dendritic piecewise linear RNN (dendPLRNN, Sect. 3.1.1) and the shallow PLRNN (shPLRNN, Sect. 3.1.2), that retain the mathematical tractability of the original PLRNN formulation [95] while achieving reconstructions in much lower-dimensional state spaces. Sect. 3.2.1 first outlines challenges encountered when training RNN models for DSR on time series observations from chaotic DS. The following sections then introduce four frameworks for training the proposed RNN models, based on sequential variational autoencoders (SVAEs, 3.2.2), including extensions to a fully probabilistic Bayesian framework, and three variants of teacher forcing (TF): sparse TF (STF, Sect. 3.2.3) generalized TF (GTF, Sect. 3.2.4) and multimodal TF (MTF, Sect. 3.2.5). Sect. 3.2.6 and Sect. 3.2.7 collect different encoder and decoder/observation models used within the SVAE and MTF approaches, while finally, Sect. 3.3 introduces a general hierarchization framework for training the proposed DSR models on time series observations collected across a group of different subjects/systems.

3.1 RNN MODELS

The network architectures introduced here build on piecewise linear RNNs (PLRNNs, [95]). PLRNNs are defined by the latent process equation

$$\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{z}_{t-1} + \boldsymbol{W}\phi(\boldsymbol{z}_{t-1}) + \boldsymbol{h} + \boldsymbol{C}\boldsymbol{s}_t, + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}). \tag{12}$$

This equation describes the evolution of an M-dimensional latent state vector $\boldsymbol{z}_t = (z_{1t} \dots z_{Mt})^T$. $\boldsymbol{A} \in \mathbb{R}^{M\times M}$ are linear self-connections, and the entries of $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ are off-diagonal nonlinear connections between units. $\boldsymbol{h} \in \mathbb{R}^M$ is a bias term, and external input $\boldsymbol{s}_t \in \mathbb{R}^K$ can be provided via $\boldsymbol{C} \in \mathbb{R}^{M\times K}$. ϕ is a nonlinear activation function, here given by the rectified linear unit (ReLU) applied element-wise:

$$\phi(\boldsymbol{z}_{t-1}) = \max(0, \boldsymbol{z}_{t-1}). \tag{13}$$

The diagonal terms in $\boldsymbol{A}$ can be interpreted as encoding different time constants of the underlying DS [355]). If trained in a probabilistic setting, such as the SVAE or MTF approach (Sect. 3.2), a Gaussian noise term $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ with diagonal covariance $\boldsymbol{\Sigma}$ is added in Eq. 12.

The PLRNN has several mathematically attractive properties due to its piecewise linear formulation. For instance, it allows the explicit computation of fixed points and cycles [104, 212, 355], and can be translated into dynamically equivalent continuous-time systems [282]. Furthermore, PLRNNs belong to the class of continuous piecewise-linear (PWL) maps, a category for which many types of bifurcations have been studied [281]. Bifurcations are fundamental in understanding how the geometrical and topological characteristics of a system's state

space are affected by its parameters. This understanding is often beneficial for characterizing or improving the training process [88, 104, 301, 350], as well as for understanding the properties of trained systems [104, 258, 259].

Additionally, the state space of the PLRNN is divided into 2^M sub-regions in which the dynamics are locally linear. Analyzing inferred PLRNNs in terms of the linear subregions inhabited by reconstructed systems leads to several interesting insights, which are discussed in more detail in Sect. 4.5.

The latent process (Eq. 12) can be connected to observations $\boldsymbol{X}$, where $(\boldsymbol{x}_t)_{t=1\dots T}$, $\boldsymbol{x}_t \in \mathbb{R}^N$ via an observation model (decoder model). While this can take many different forms, collected in Sect. 3.2.7, in the simplest case, we can assume a linear Gaussian observation model:

$$\boldsymbol{x}_t = \boldsymbol{B}\boldsymbol{z}_t + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}). \tag{14}$$

Here $\boldsymbol{B} \in \mathbb{R}^{N\times M}$ is a factor loading matrix and $\boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$ Gaussian observation noise with diagonal covariance $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times N}$. As with the latent process noise $\boldsymbol{\Sigma}$, the covariances are only explicitly estimated in the SVAE and MTF approach.

ON THE RELATIONSHIP BETWEEN LATENT DIMENSION AND OBSERVATION DIMENSION Note that the observation model N can in principle both imply that the latent states project to higher dimensional ($M < N$) or lower-dimensional observations ($M > N$) (see also Sect. 2.3). In neuroscience, state space models are often used as nonlinear dimensionality reduction tools for extracting lower-dimensional manifolds formed by a system's attractors from high-dimensional observations. This is particularly the case if there is redundancy between dimensions of the observed time series, as hypothesized for neural activity in parts of the brain [225]. For the Electroencephalography (EEG) data in [152], this allowed us to reduce the reconstructions from 64 observed channels to a 16-dimensional state space (see Sect. 4.2.2). More generally, the observation that many high-dimensional real-world datasets (such as images) lie on relatively low-dimensional latent manifolds is known as the 'Manifold' hypothesis [111] and is more widely discussed in ML. Many popular ML techniques, such as variational autoencoders (see Sect. 3.2.2) or CNNs [290] can be viewed as implicitly or explicitly serving as nonlinear dimensionality techniques, contributing an explanation for the surprising generalization abilities of ML algorithms [30, 360].

However, for time series measurements of latent DS, we often face the reverse problem when not all the system's dimensions are observed. As discussed in the introduction, in this case, systems can be embedded in higher-dimensional state spaces, e.g. by using temporal delay embeddings ([21, 187, 349], Fig. 4). For the electrocardiogram (ECG) data (Sect. 4.2.2), which was provided as a one-dimensional electrophysiological time series, we, in turn, delay-embedded the system into a 5-dimensional latent space. The required latent dimension also depends on the RNN architecture and can play a similar role as the width of classical NNs [256] in determining its expressivity. The two RNN models proposed in the following were developed to enhance the expressivity of the PLRNN (Eq. 12) while retaining its tractability, thus enabling reconstructions in lower-dimensional latent spaces.

3.1.1 *Dendritic PLRNN*

DENDRITIC COMPUTATION AND SPLINE BASIS EXPANSION Dendrites are believed to significantly contribute to neural computation [205, 271, 272], e.g. by the amplification of synaptic inputs to neurons [166, 354]. The concept of dendritic branches functioning as semi-independent computational units has also been likened to the computation of a 2-layer neural network [270, 271, 316, 317]. Inspired by these principles, we integrate dendritic processing into the latent equation of the PLRNN, Eq.12, by employing a linear mix of ReLU-type threshold nonlinearities [54], as depicted in Fig. 7. This is achieved by modifying Eq. 13 to:

$$\phi(z_{t-1}) = \sum_{b=1}^{B} \alpha_b \max(0, z_{t-1} - h_b). \tag{15}$$

Here we define the dendritic input/output slopes $\alpha_b \in \mathbb{R}$ and activation thresholds $h_b \in \mathbb{R}^M$, and B is the number of dendrites. Mirroring real dendrites, which adjust their morphology when learning takes place [317, 378], these parameters are jointly optimized with the other PLRNN parameters. The system encompassing Eqs. 12, and 15 is called the *dendPLRNN* [54].

When Eq. 15 is integrated into Eq. 12, it represents a linear spline basis expansion. Such expansions are well-studied in statistics and ML for function approximation [147, 380, 412] in regression and model-based scenarios.

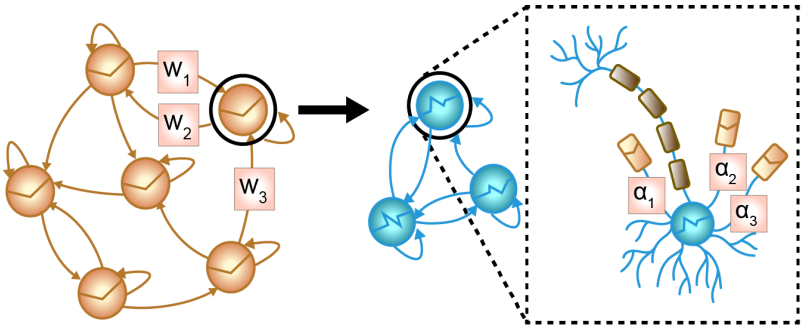

Figure 7: The dendPLRNN extends each neuron of the PLRNN into a set of nonlinear branches, significantly increasing its expressivity and enabling reconstructions in lower dimensions. Figure created with the artistic support of Darshana Kalita. Taken from [54].

MATHEMATICAL TRACTABILITY AND DYNAMICAL SYSTEMS INTERPRETATION Sharp threshold nonlinearities, such as the ReLU function, align well with neurobiological perspectives, as dendrites exhibit similar threshold-type behavior [205, 272]. More importantly, this choice preserves the theoretical properties of the PLRNN, including its mathematical tractability, which allows analytical access to fixed points and cycles [95, 104], as discussed previously. For example, for a dendPLRNN trained on the Lorenz-63 system (Fig. 21), I computed all fixed points in under 1 second and cycles up to the 40^{th} order within 20 seconds using a single 1.8GHz CPU. The tractability of the dendPLRNN naturally follows from the proposition that any dendPLRNN can be rewritten as a conventional PLRNN [54]:

Theorem 1 *A dendPLRNN of* M *dimensions can always be restructured as a standard PLRNN with dimensions* $M \times B$*, represented as*

$$\hat{\boldsymbol{z}}_t = \tilde{\boldsymbol{A}}\hat{\boldsymbol{z}}_{t-1} + \tilde{\boldsymbol{W}}\max(0, \hat{\boldsymbol{z}}_{t-1}) + \hat{\boldsymbol{h}}_0 + \tilde{\boldsymbol{C}}\boldsymbol{s}_t + \tilde{\boldsymbol{\epsilon}}_t. \tag{16}$$

The detailed proof is given in [54]. This theorem also provides some intuition on how a dendPLRNN can reduce the dimensionality of a reconstructed system since a high-dimensional PLRNN could often be represented as a lower-dimensional yet equally effective dendPLRNN. Fig. 8 illustrates that a dendPLRNN enables reconstructions in much lower dimensions than PLRNNs. Details on the precise computation of fixed points and k-cycles for the dendPLRNN are provided in [54].

A notable issue with PLRNNs is the potential unboundedness of latent states due to the ReLU function. The dendPLRNN, however, provides a straightforward and natural solution to limit latent states while retaining the piecewise linear formulation:

Theorem 2 *For each basis* $\{\alpha_b, \boldsymbol{h}_b\}$ *in Eq. 15 of a dendPLRNN, we can add another basis* $\{\alpha_b^*, \boldsymbol{h}_b^*\}$ *with* $\alpha_b^* = -\alpha_b$ *and* $\boldsymbol{h}_b^* = \boldsymbol{0}$*. Then, for* $\sigma_{\max}(\boldsymbol{A}) < 1$*, any orbit of this clipped dendPLRNN (Eq. 17) will remain bounded.*

The proof for this theorem is likewise given in [54]. The clipped dendPLRNN model is then defined by:

$$\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{z}_{t-1} + \boldsymbol{W}\sum_{b=1}^{B}\alpha_b\left[\max(0, \boldsymbol{z}_{t-1} - \boldsymbol{h}_b) - \max(0, \boldsymbol{z}_{t-1})\right] + \boldsymbol{h}_0. \tag{17}$$

3.1.2 *Shallow PLRNN*

A further advance in formulating low-dimensional, mathematically tractable RNN models is the *shallow* PLRNN (shPLRNN), introduced in [152] as:

$$\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{z}_{t-1} + \boldsymbol{W}_1\max(0, \boldsymbol{W}_2\boldsymbol{z}_{t-1} + \boldsymbol{h}_2) + \boldsymbol{h}_1, \tag{18}$$

with latent states $\boldsymbol{z}_t \in \mathbb{R}^M$, diagonal matrix $\boldsymbol{A} \in \mathbb{R}^{M\times M}$, rectangular connectivity matrices $\boldsymbol{W}_1 \in \mathbb{R}^{M\times L}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{L\times M}$, and thresholds $\boldsymbol{h}_2 \in \mathbb{R}^L$ and $\boldsymbol{h}_1 \in \mathbb{R}^M$. Here, L is the dimension of the hidden layer. With $L > M$, the network expands each unit's activation into a weighted sum of ReLU nonlinearities.

The shPLRNN can be rewritten in the form of a dendPLRNN. It follows, in particular, that fixed points and cycles of Eq. 18 can be computed analogously as for the dendPLRNN. Vice versa, any M-dimensional dendPLRNN can be reformulated as an M-dimensional shPLRNN with hidden layer size $L = M \cdot B$. The detailed proof is given in [152]. As for the dendPLRNN, the shPLRNN naturally incorporates a clipping mechanism that prevents state divergence provided the largest absolute eigenvalue of $\boldsymbol{A}$ is smaller than 1 [152]:

$$\begin{aligned}\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{z}_{t-1} + \boldsymbol{W}_1\big[&\max(0, \boldsymbol{W}_2\boldsymbol{z}_{t-1} + \boldsymbol{h}_2) \\ &- \max(0, \boldsymbol{W}_2\boldsymbol{z}_{t-1})\big] + \boldsymbol{h}_1.\end{aligned} \tag{19}$$

The shPLRNN enables reconstructions of systems such as the Lorenz-63 or Lorenz-96 system directly in the observation dimension ($M = N$) (see Table

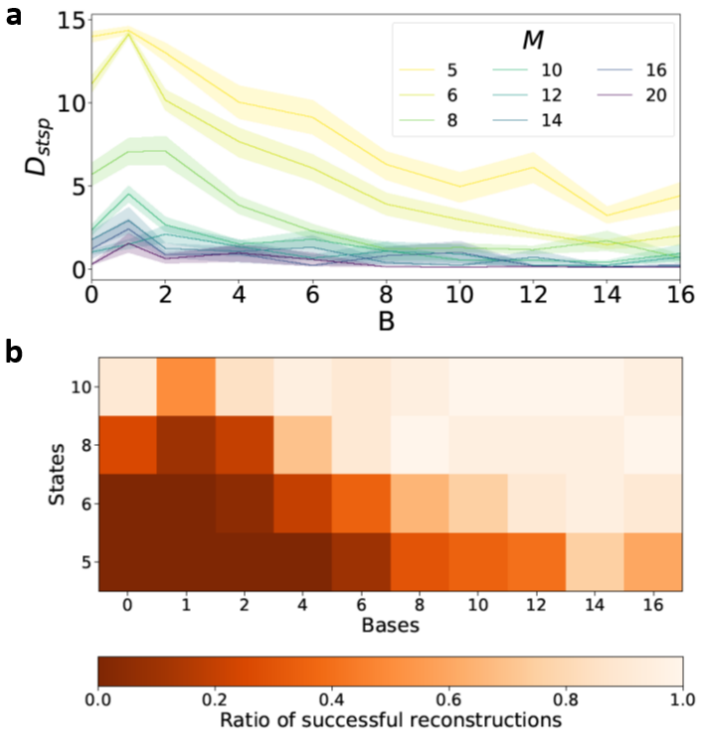

Figure 8: Basis expansion reduces latent space dimensionality. **a**: Agreement in attractor geometries (Sect. 4.1) (top) and proportion of successful reconstructions (bottom) for the Lorenz-63 system as a function of the number of bases (B) and latent states (M). $B = 0$ in the top graph denotes the standard PLRNN (no basis expansion). **b**: Runs with $D_{stsp} < 4$ were defined as successful (similar results are obtained with other choices for the D_{stsp} threshold). Taken from [54].

7), a result not possible both with the PLRNN and dendPLRNN, using the same training techniques (see also Fig. 8). This observation indicates that the specific form of the nonlinearity, which is applied in the usually much higher-dimensional hidden space $L >> M$ of the shPLRNN via $\max(\boldsymbol{W}_2 \boldsymbol{z}_{t-1} + \boldsymbol{h}_2)$, is particularly powerful at reducing the number of units M required for successful reconstructions with the RNN model.

Despite the implementation of a clipping mechanism (Eqs. 17 and 19) for the two RNN models, divergences can still occur if entries of the linear self-connections matrix $\boldsymbol{A}$ exceed one, $\sigma_{max}(\boldsymbol{A}) > 1$. However, these divergences can generally be mitigated by incorporating additional regularizations during training. One possibility is to penalize the self-connections that approach a threshold around one too closely:

$$\mathcal{L}_A = \lambda_A \sum_i \max(0, a_{ii} - \theta)^2$$

where λ_A is a regularization coefficient, a_{ii} are the entries of the diagonal matrix $\boldsymbol{A}$, and θ is a threshold value slightly less than one (e.g. ≈ 0.995). Additionally, regularizing the magnitude of the latent states $\boldsymbol{z}_t$ has also often proven effective in preventing divergences:

$$\mathcal{L}_{latent} = \lambda_{latent} \sum_t \|\boldsymbol{z}_t\|^2$$

where again λ_{latent} is a regularization constant. While divergences were generally not an issue on synthetic benchmark systems, including these regularizations was helpful on some of the real-world datasets (Sect. 4.3.3) or for the reconstructions from symbolic DS (Sect. 4.3.2). Fig. 13 further outlines how different training methods can influence the sensitivity of the loss landscape to divergences in the DSR model.

Finally, as proposed in Schmidt et al. [355], a manifold attractor regularization (MAR) can be added to the loss function on a subset of states $M_{\text{reg}} \leqslant M$ to encourage the discovery of long-term dependencies and slow time scales in the data:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \left(\sum_{i=1}^{M_{\text{reg}}} (A_{ii} - 1)^2 + \sum_{i=1}^{M_{\text{reg}}} \sum_{j \neq i}^{M} (W_{ij})^2 + \sum_{i=1}^{M_{\text{reg}}} h_i^2 \right). \tag{20}$$

This regularization pushes a subset of states towards a stable manifold of equilibrium points and can be included for both the PLRNN and the dendPLRNN. For the shPLRNN, in [152] we adjusted this regularization to

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \left(\|\mathbb{1} - \boldsymbol{A}\|_F^2 + \|\boldsymbol{W}_1\|_F^2 + \|\boldsymbol{W}_2\|_F^2 + \|\boldsymbol{h}_1\|_2^2 + \|\boldsymbol{h}_2\|_2^2 \right) \tag{21}$$

where $\|\|_F$ denotes the Frobenius norm. The MAR can aid with the learning of systems with different time scales, such as the bursting neuron model ([94, 355]). The MAR was used when training the RNN models within the SVAE approach (Sect. 3.2.2) for the results in Sect. 4.2.1, and helps with stabilizing estimates for the annealing protocol for GTF [152].

3.2 TRAINING ALGORITHMS

3.2.1 *Challenges of Training RNN Models on Chaotic Dynamical Systems*

Before delving into specific training algorithms, I will provide a brief overview of the theoretical results from Mikhaeil, Monfared, and Durstewitz [276] for understanding the challenges of training RNNs when modeling chaotic systems with gradient-based methods, following the outline in [152].

An RNN model with parameters $\boldsymbol{\theta}$ constitutes a recursive map as defined in Eq. 3:

$$\boldsymbol{z}_t = \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}, \boldsymbol{s}_t), \tag{22}$$

where the $\boldsymbol{s}_t$ denote external inputs. Note that this definition does not make any assumptions about the specific parameterization of the RNN, and holds for the dendPLRNN or shPLRNN as well as other popular RNNs such as Long Short Term Memory (LSTM) networks [155].

Training RNNs often requires sampling sequences from the RNN, and hence the iterative application of Eq. 22. The to-date most popular algorithms for training RNNs are based on variants of Backpropagation Through Time (BPTT; [340, 420]), where RNN sequences of length T are drawn and gradients are propagated backward in time through the network. The computation of these gradients rests on a loss function $\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}_t(\boldsymbol{z}_t, \bar{\boldsymbol{z}}_t)$, where $\mathcal{L}_t$ is often just the mean-squared error, and with $\boldsymbol{z}_t$ being the outputs of the RNN and $\bar{\boldsymbol{z}}_t$ the

targets. During training, the gradients of the RNN parameters $\theta_i \in \boldsymbol{\theta}$ are then obtained by unfolding the RNN over time:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial \theta_i}, \quad \text{with} \quad \frac{\partial \mathcal{L}_t}{\partial \theta_i} = \sum_{r=1}^{t} \frac{\partial \mathcal{L}_t}{\partial z_t} \frac{\partial z_t}{\partial z_r} \frac{\partial^+ z_r}{\partial \theta_i}, \tag{23}$$

It has been long observed that the repeated application of recurrent connections during the backward pass of gradients in time can lead to exploding and vanishing gradient problems (EVGP, [33]). Many RNN architectures, such as LSTMs [155] or, more recently Long Expressive Memories (LEM; [341]), have been designed to deal with the EVGP explicitly.

However, the central insight from [276] is that when training via BPTT on time series observations from a chaotic DS, exploding gradients are in principle unavoidable. To see this, first note that RNN models become surrogates of the underlying ground truth system during training, and an RNN modeling a chaotic DS likewise has to constitute a chaotic DS. As detailed in 2.2, chaotic DS are characterized by trajectory divergence, formally encapsulated by the system's maximum Lyapunov spectrum (Eq. 5). The maximum Lyapunov exponent of an RNN orbit $\boldsymbol{Z} = \{z_1, z_2, \ldots, z_T, \ldots\}$ is given by the product of Jacobians $\boldsymbol{J}_t$ along the orbit by

$$\lambda_{max} := \lim_{T\to\infty} \frac{1}{T} \log \left\| \prod_{r=0}^{T-2} \boldsymbol{J}_{T-r} \right\|_2, \tag{24}$$

where $\|\cdot\|_2$ denotes the spectral norm, and the Jacobians are given by

$$\boldsymbol{J}_t := \frac{\partial \boldsymbol{F}_\theta(z_{t-1}, s_t)}{\partial z_{t-1}} = \frac{\partial z_t}{\partial z_{t-1}}, \tag{25}$$

This same product is present in the gradients of the loss function in Eq. 23, which can be seen more directly when rewriting the derivatives of two states $\frac{\partial z_{t_2}}{\partial z_{t_1}}$ at times t_1 and t_2 (with $t_2 > t_1$) as:

$$\begin{aligned} \frac{\partial z_{t_2}}{\partial z_{t_1}} &= \frac{\partial z_{t_2}}{\partial z_{t_2-1}} \frac{\partial z_{t_2-1}}{\partial z_{t_2-2}} \cdots \frac{\partial z_{t_1+1}}{\partial z_{t_1}} \\ &= \boldsymbol{J}_{t_2} \boldsymbol{J}_{t_2-1} \cdots \boldsymbol{J}_{t_1+1} = \prod_{k=0}^{t_2-t_1-1} \boldsymbol{J}_{t_2-k}, \end{aligned} \tag{26}$$

Therefore, the gradients in BPTT (Eq. 23) contain the same product of Jacobians also present in the computation of the maximum Lyapunov exponent in Eq. 24. Mikhaeil, Monfared, and Durstewitz [276] prove that this leads to exponentially increasing loss gradients as $T \to \infty$. In practical scenarios, this results in unstable and challenging training even for moderate sequence lengths T (see also Fig. 48). On the other hand, capturing slow time scales of the underlying system often requires training on longer sequences, and we have observed that long training sequences particularly help when training on experimental datasets [54, 152]. Additionally, even with more complex architectures like LSTMs [155] or LEMs [341] that are designed to manage gradient flows and mitigate the EVGP, or by straightforward gradient clipping methods, the underlying issue remains unresolved [276]. The training methods described in the rest of this thesis all deal with the challenge posed by this insight implicitly

or explicitly. Training with SVAEs avoids the problem altogether by only training on 1-step ahead predictions and not sampling longer sequences from the DSR model. Sparse, generalized, and multimodal TF directly tackle gradient divergence by providing the model with control-theoretic forcing signals that pull diverging trajectories 'back on track' during training.

3.2.2 *Sequential Variational Autoencoders (SVAEs)*

GENERATIVE MODELS OF TIME SERIES The DSR models introduced in the previous section can be framed in the language of probabilistic, generative latent variable models outlined in Sect. 2.3. Given the assumptions on the factorization of the probabilistic graph (Eqs. 10 and 11), the PLRNN models introduced in the previous section are naturally expressed as a probabilistic generative time series model by writing the latent model and observation model as probability densities according to:

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) = \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_{z_t}, \Sigma), \tag{27}$$

and likewise for the observations:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{x_t}(\boldsymbol{z}_t), \Gamma). \tag{28}$$

This relationship is e.g. given by the linear Gaussian observation model (Eq. 14). More complex observation models, such as a hemodynamic response function for fMRI data [212], can also depend on multiple latent states across a time window τ. In the case of non-Gaussian observations like count or ordinal data, the latent states are linked to decoder models that reflect the discrete nature of the observations, using e.g. generalized linear models (see Sect. 3.2.7).

VARIATIONAL INFERENCE Variational autoencoders (VAEs) have since their inception [202, 330] gained widespread traction for the training of deep generative latent variable models, particularly after the introduction of the reparameterization trick [202, 330], enabling their efficient training with gradient-based methods. VAEs are encoder/decoder architectures that are trained to optimize the Evidence Lower Bound (ELBO), a concept closely linked to the Helmholtz free energy, and already used for training Helmholtz machines [83]. The main idea behind VAEs is to introduce a variational density q over the latent states $\boldsymbol{Z}$ which is generally approximated from the data directly by some trainable NN architecture with parameters ϕ, such as RNNs, Convolutional NNs or Transformers (see Sect. 3.2.6). Introducing this approximate posterior leads to an expression that is easier to optimize with gradient-based methods than the true posterior, which is usually intractable since no closed-form solutions exist and its computation requires evaluating high-dimensional integrals [44]. The ELBO constitutes a lower bound on the data likelihood, derived using Jensen's inequality (see e.g. [131] for a detailed derivation):

$$\begin{aligned} \log p_\theta(\boldsymbol{X}) &\geqslant \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\boldsymbol{X}, \boldsymbol{Z})}{q_\phi(\boldsymbol{Z}|\boldsymbol{X})}\right] \\ &= \mathbb{E}_{q_\phi}\left[\log p_\theta(\boldsymbol{X}, \boldsymbol{Z})\right] - \mathbb{E}_{q_\phi}\left[\log q_\phi(\boldsymbol{Z}|\boldsymbol{X})\right] \\ &= -\mathrm{KL}\left(q_\phi(\boldsymbol{Z}|\boldsymbol{X}) \| p_\theta(\boldsymbol{Z}|\boldsymbol{X})\right) + \log p_\theta(\boldsymbol{X}) \\ &= \mathrm{ELBO}_{\boldsymbol{X}}(\phi, \theta). \end{aligned} \tag{29}$$

The KL divergence term in the third line measures the dissimilarity between the approximate posterior and the true posterior distribution. This divergence is always non-negative and only equals zero if the two distributions are identical. If the approximate posterior perfectly captures the true posterior, the ELBO is, therefore, equal to the log-likelihood of the data $\log p_\theta(\boldsymbol{X})$, reaching the upper bound of the ELBO.

During training with VI, the expectation value $\mathbb{E}_{q_\phi}$ is approximated using Monte Carlo samples from the approximate posterior:

$$\text{ELBO}_{\boldsymbol{X}}(\phi, \boldsymbol{\theta}) \approx \frac{1}{L} \sum_{\ell=1}^{L} \left[\log p_\theta(\boldsymbol{X}, \boldsymbol{Z}^{(\ell)}) - \log q_\phi(\boldsymbol{Z}^{(\ell)}|\boldsymbol{X}) \right]. \tag{30}$$

Sampling directly from the approximate posterior $q_\phi(\boldsymbol{Z}|\boldsymbol{X})$ can introduce high variance into the gradients of the ELBO with respect to the parameters ϕ. The reparameterization trick overcomes this by expressing the random variable $\boldsymbol{Z}$ as a deterministic transformation of a fixed distribution ϵ (usually chosen to be standard normal) and parameters ϕ:

$$\boldsymbol{Z}^{(\ell)} = g_\phi(\boldsymbol{\epsilon}^{(\ell)}, \boldsymbol{X}), \quad \boldsymbol{\epsilon}^{(\ell)} \sim p(\boldsymbol{\epsilon}), \tag{31}$$

Several state-of-the-art generative modeling approaches [314], such as normalizing flows [331] and diffusion models [371], build on the idea of combining simple fixed noise distributions with flexible trainable (invertible) transformations for optimizing complex statistical models.

Training RNNs with VI is conceptually illustrated in Fig. 9 (a more detailed treatment is given e.g. in [351]), and usually referred to as training a sequential variational autoencoder (SVAE). Spelling out the first term in the ELBO (Eq. 29) explicitly, one obtains a joint likelihood over latent states and observations that factorizes according to the assumptions from Eqs. 27 and 28:

$$p(\boldsymbol{X}, \boldsymbol{Z}) = p(\boldsymbol{z}_1) p(\boldsymbol{x}_1|\boldsymbol{z}_1) \prod_{t=2}^{T} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) p(\boldsymbol{x}_t|\boldsymbol{z}_t). \tag{32}$$

Spelling out the joint log-likelihood over the M-dimensional latent states and N-dimensional observations explicitly leads to:

$$\begin{aligned} \log p(\boldsymbol{X}, \boldsymbol{Z}) = &-\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_0| - \frac{1}{2}(\boldsymbol{z}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{z}_1 - \boldsymbol{\mu}_0) \\ &-\frac{N}{2}\log(2\pi) - \frac{1}{2}\log\boldsymbol{\Gamma}| - \frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{\mu}_{x_1})^T \boldsymbol{\Gamma}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_{x_1}) \\ &+ \sum_{t=2}^{T} \left(-\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{z}_t - \boldsymbol{\mu}_{z_t})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_t - \boldsymbol{\mu}_{z_t}) \right) \\ &+ \left(-\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_{x_t})^T \boldsymbol{\Gamma}^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_{x_t}) \right). \end{aligned} \tag{33}$$

The initial state and its covariance are usually taken as free model parameters and jointly optimized during training. However, they can also be estimated from the data directly using the encoder model. This joint likelihood is the first term in the second line in Eq. 29. What remains for the computation of the ELBO is the expectation over the approximate posterior density, which is just its entropy $\mathbb{H}_{q_\phi}$. This entropy can often be analytically computed given certain simplifying assumptions about the parameterization of the approximate posterior.

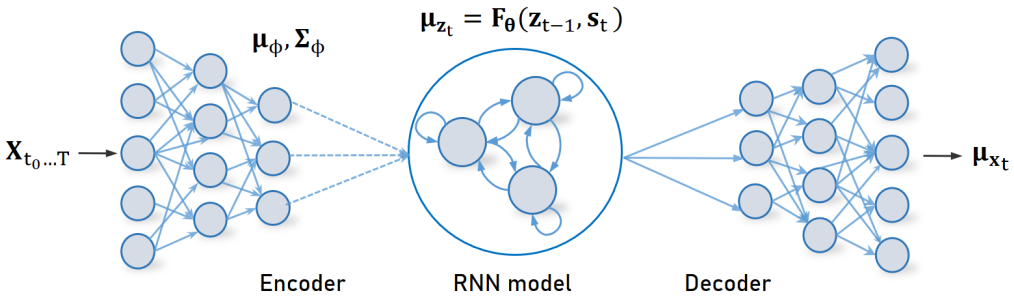

Figure 9: Illustration of the SVAE setup.

APPROXIMATE POSTERIOR Training the SVAE means maximizing the ELBO, which, in turn, implies maximizing the likelihood of the data $\boldsymbol{X}$. The most important practical choice lies in choosing an appropriate parameterization for the approximate posterior via the encoder model $q_\phi(\boldsymbol{Z}|\boldsymbol{X})$. The approximate posterior is usually assumed to be a multivariate Gaussian of the form [44]:

$$q_\phi(\boldsymbol{Z}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{X}), \boldsymbol{\Sigma}_\phi(\boldsymbol{X})) \tag{34}$$

Here, the mean and covariance are functions of the observations. Assuming a trajectory of observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ and corresponding latent states $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T\}$ of length T, the main challenge is to find a formulation of the approximate posterior that both captures the complexity and temporal structure of the underlying time series while making the computation of the entropy in the ELBO (Eq. 30) resource-efficient. Computing the entropy requires evaluating the determinant of the covariance matrix $\boldsymbol{\Sigma}_{\phi(\boldsymbol{x})} \in \mathbb{R}^{MT \times MT}$, which naively scales with $\mathcal{O}(T^2)$, and thus becomes costly to evaluate for longer sequence lengths T. Thus for computational efficiency, for the results in [54] and [53], we assumed a mean-field approach, breaking down $q_\phi(\boldsymbol{Z}|\boldsymbol{X})$ over time by introducing a time-varying mean $\boldsymbol{\mu}_{t,\phi}$ and covariance $\boldsymbol{\Sigma}_{t,\phi}$ at every time step, and further assuming the covariance matrix is diagonal. Other approximations based on a block-tri-diagonal covariance structure that mimic the assumptions of a first-order Markov process have been proposed in Archer et al. [16], and employed in publications in our group in Kramer et al. [214], and as a comparison in [53]. However, the Cholesky factorization required in the computation of the determinant of the covariance matrix becomes a significant computational bottleneck. This makes a full mean-field approximation computationally desirable while constituting a potentially overly simplifying assumption [26, 44]. For this thesis, I have implemented and tested a number of encoder models, detailed in Sect. 3.2.6, including direct comparisons in Table 2. A parameterization based on temporal convolutional neural networks (CNNs) [227] was most successful for the experiments in [53, 54].

PROBLEMS WITH SVAES IN DSR Given the mean-field assumption, the joint likelihood and the entropy calculations become fully parallelizable. Combined with encoder models based on architectures that likewise scale well on long sequences, such as CNNs, this renders the runtime of the SVAE almost independent of the sequence length T. Training within the SVAE approach does not require drawing sequences from the DSR model (Eq. 22) during training since the joint likelihood only contains one-step ahead predictions (Eq. 33), avoiding the exploding gradients discussed in the Sect. 3.2.1. The burden for capturing

long-term temporal dependencies is hence directly put on the encoder model. However, extensive experiments in the context of [54] and comparisons with the results from [214] revealed clear problems with this setup. Even when relying on sophisticated encoder models, implementing the block tri-diagonal covariance structure based on the Markovian latent model assumption [16, 214], the results on anything but simple benchmark systems like the Lorenz-63 system were sub-par, and reconstructions on challenging experimental datasets were next to impossible [54, 214]. This observation is more directly illustrated in Fig. 13, which outlines how training an SVAE in the context of DSR leads to overly smoothened loss landscapes ill-equipped to correctly capture long-term statistics of the reconstructed system. These results, combined with the much better performance of TF-based techniques in achieving this goal (also illustrated in Fig. 13), motivated the development of the MTF framework. This approach combines the advantages of the SVAE, such as its flexible encoder-decoder structure and its potential to integrate across multiple modalities, with the strong performance of TF-based training in DSR.

BAYESIAN DATA INTEGRATION In the SVAE, latent states $\boldsymbol{Z}$ are treated as random variables, but model parameters are not. A natural extension of the SVAE into a fully probabilistic setting is to treat all or a subset of the model parameters of the DSR model (e.g. $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{W}, \boldsymbol{h}, \boldsymbol{B}, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}\}$ for the PLRNN in Eq. 12) as random variables as well. Similar ideas can be traced back to Bayesian NNs [306]. Bayesian approaches for RNNs also bear some resemblance to concepts in modeling stochastic dynamics, such as random DS [17] or random ODEs (RDEs, [90]). Here model parameters themselves are seen as stochastic components of a DS. RDEs are conjugate to SDEs [144, 171], and hence stochasticity in model parameters is intimately tied to stochasticity in the modeled dynamics.

A fully Bayesian approach was implemented and tested within the SVAE framework in our group jointly with the work in Sayer [351], whose notation I will follow here. Assume we have observed time series $\boldsymbol{X}$ from some DS as before, but also observed some other structural data $\boldsymbol{D}$ that could contain prior knowledge about the system. Treating model parameters as random variables leads to a joint posterior over $\boldsymbol{Z}$ and model parameters $\boldsymbol{\theta}$, which can be written as a product:

$$p(\boldsymbol{Z}, \boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{D}) = p(\boldsymbol{Z} \mid \boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{D}) p(\boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{D}).$$

The posterior over model parameters can be spelled out explicitly using Bayes' formula:

$$p(\boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{D}) = \frac{p(\boldsymbol{X} \mid \boldsymbol{\theta}, \boldsymbol{D}) p_\alpha(\boldsymbol{\theta} \mid \boldsymbol{D})}{p(\boldsymbol{X} \mid \boldsymbol{D})},$$

This approach therefore introduces a prior distribution $p_\alpha = (\boldsymbol{\theta}|\boldsymbol{D})$ over model parameters. This prior provides a natural way to integrate data/prior knowledge $\boldsymbol{D}$ into the model, which might often be available in empirical settings (e.g. neural structural information or psychological survey data). It further incorporates two approximate posterior distributions, one over model parameters and one over latent states:

$$q_\xi(\boldsymbol{Z}, \boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{D}) = q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{D}) q_\psi(\boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{D}).$$

Deriving an optimization criterion based on these distributions is somewhat lengthy (see [351] for more details). The resulting optimization criterion however is relatively intuitive:

$$\mathbb{L}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\alpha}) = \Big\{ \mathbb{E}_{q_{\boldsymbol{\phi}}} \Big[\mathbb{E}_{q_{\boldsymbol{\psi}}} \log p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta}, \boldsymbol{D}) \Big] - \mathrm{KL}\Big(q_{\boldsymbol{\psi}}(\boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{D}) \parallel p_{\boldsymbol{\alpha}}(\boldsymbol{\theta} \mid \boldsymbol{D}) \Big) + \mathbb{H}\Big(q_{\boldsymbol{\phi}}(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{D}) \Big) \Big\}.$$

This term consists of the joint likelihood (Eq. 33) and entropy over the approximate posterior over latent states (the ELBO in the SVAE), combined with a KL divergence between approximate posterior and prior over the model parameters. This expression contains expectation values over both approximate posteriors, which can again be approximated by Monte-Carlo sampling using the reparameterization trick:

$$\mathbb{L}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\alpha}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{L} \sum_{\ell=1}^{L} \Big(\log p(\boldsymbol{X}, \boldsymbol{Z}^{(\ell)} | \boldsymbol{\theta}^{(k)}, \boldsymbol{D}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}^{(k)} | \boldsymbol{D}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{Z}^{(\ell)} | \boldsymbol{X}, \boldsymbol{D}) - \log q_{\boldsymbol{\psi}}(\boldsymbol{\theta}^{(k)} | \boldsymbol{X}, \boldsymbol{D}) \Big), \tag{35}$$

While this expression can be used as an optimization criterion for training a PLRNN on time series data $\boldsymbol{X}$ and structural data $\boldsymbol{D}$, the crux in successfully using this framework for DSR lies in designing appropriate prior $p_{\boldsymbol{\alpha}}$ and approximate posterior distributions $q_{\boldsymbol{\phi}}$ over latent states and $q_{\boldsymbol{\psi}}$ over the model parameters. In both cases, we can assume multivariate normal distributions parameterized by NNs, leading to analytical expressions for the likelihoods in Eq. 35. We tested different architectures to parameterize the respective distributions, such as fully connected NNs for $p_{\boldsymbol{\alpha}}$ or LSTMs for $q_{\boldsymbol{\psi}}$. Within this framework, it however proved challenging to achieve successful DSR even for simple benchmark systems. Some of these challenges can be tied to the overall inferiority of the SVAE approach explained above, since the optimization criterion for the latent states remains the same as in Eq. 33. Further, taking model parameters $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{W}\}$ as random variables likely destabilizes training, since as observed in [104, 149], often small changes in the parameters of the PLRNN can lead to large qualitative changes in dynamics. Since model parameters are drawn during training, this would be reflected in a high variance of the error gradients given different samples from $q_{\boldsymbol{\psi}}$ or jumps in the loss landscape around bifurcation boundaries, as detailed in [104]. Thus, although theoretically appealing and offering a natural method to incorporate structural information $\boldsymbol{D}$, an approach successfully leveraging Bayesian integration in DSR requires further research.

3.2.3 *Sparse Teacher Forcing (STF)*

The three TF approaches described in the following sections directly address the main challenge framed in Sect. 3.2.1: how can we avoid diverging gradients on chaotic systems and still capture long-term statistics of the underlying system correctly by sampling long RNN trajectories during training? The main idea behind TF is to leverage a combination of forward-iterated latent states (those

potentially incurring exploding gradients during the backward pass when iterating Eq. 22) and data-inferred states (estimated in some way from the data) to balance the training process by 'forcing' trajectories back on track. While TF has been discussed in the literature before [178, 307, 310, 421], the connections to chaotic dynamics and successful applications in the context of DSR has only been thoroughly explored in our group in the works discussed here [53, 54, 152], and in [276].

SPARSE TF In sparse TF (STF), we directly replace latent states (or a subset of them) with states inferred from observations at intervals τ while leaving the network to evolve freely otherwise. What remains to be determined is the relationship between observations and latent states, reminiscent of what is learned by the encoder model in the SVAE approach. Since the STF approach combines classical RNN training with BPTT with TF, it is also called 'BPTT-TF' in [54].

If we assume observations to be normally distributed, as when employing the observation model from Eq. 14, then the reverse direction of the observation equation is obtained by building the Moore–Penrose (pseudo-) inverse of $\boldsymbol{B}$, taking $\hat{\boldsymbol{z}}_t = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{x}_t$. When using the full observation model, $\boldsymbol{B}$ occasionally becomes ill-conditioned (close to singular) during training. To avoid this, we can regularize the condition number of $\boldsymbol{B}$ [152], ensuring invertibility:

$$\mathcal{L}_{cn} = \lambda_{cn}\left(1 - \frac{\sigma_{max}(\boldsymbol{B})}{\sigma_{min}(\boldsymbol{B}) + \epsilon}\right)^2. \tag{36}$$

Here $\sigma_{max}(\boldsymbol{B})$ and $\sigma_{min}(\boldsymbol{B})$ are the largest and smallest singular values of $\boldsymbol{B}$, respectively, λ_{cn} is a regularization constant and $\epsilon = 10^{-8}$ is a small number added for numerical stability.

IDENTITY TF Instead of training a linear matrix $\boldsymbol{B}$, the inversion becomes trivial if we instead adopt an identity mapping as the observation model:

$$\hat{\boldsymbol{x}}_t = \mathcal{I}\boldsymbol{z}_t, \tag{37}$$

with $\mathcal{I} \in \mathbb{R}^{N\times M}$, and an identity matrix with $\mathcal{I}_{kk} = 1$ for the k read-out neurons, $k \leqslant N$, and zeroes elsewhere. This training technique is called identity-TF (id-TF). Since the observation model boils down to an identity mapping between observations and latent states, the read-out neuron states can be directly replaced with observations at every τ time steps:

$$\boldsymbol{z}_{t+1} = \begin{cases} \mathrm{RNN}(\tilde{\boldsymbol{z}}_t) & \text{if } t \in \mathcal{F} \\ \mathrm{RNN}(\boldsymbol{z}_t) & \text{else} \end{cases} \tag{38}$$

with $\mathcal{F} = \{l\tau + 1\}_{l\in\mathbb{N}_0}$, and where we assume that teacher-forced states are equal to the observations, $\tilde{\boldsymbol{z}}_t = \boldsymbol{x}_t$. Training with id-TF takes place in a fully deterministic framework where latent states only factor into the training objective indirectly through the observations, and hence the observation model in Eq. 37 does not contain any parameter for the noise covariance. The more complex likelihood function of the SVAE in Eq. 33 over both latent states and observations then reduces to the Mean Squared Error (MSE) loss function over model predictions and observations:

$$\ell_{MSE}(\hat{\boldsymbol{X}}, \boldsymbol{X}) = \frac{1}{N\cdot T}\sum_{t=1}^{T}\|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2^2, \tag{39}$$

where $\hat{\boldsymbol{X}}$ represents the model predictions and $\boldsymbol{X}$ denotes the training sequence of length T. The $M - k$ non-readout neurons do not contribute to the loss function but help to embed the approximated DS into a higher-dimensional space that increases expressivity and can help to represent unobserved variables of the underlying DS (as illustrated e.g. in Chapter 5).

Finding an optimal TF interval τ is a choice of much practical importance, and is often instrumental in achieving successful DSR. [276] suggest selecting τ based on the predictability time, defined as

$$\tau_{pred} = \frac{\ln 2}{\lambda_{max}}. \tag{40}$$

where λ_{max} is the maximum Lyapunov exponent of the underlying system. When no ground-truth value for λ_{max} exists, such as when training on experimental data, it needs to be estimated numerically from the data, e.g. using the Julia library `DynamicalSystems.jl` [82]. However, in most practical settings, such as for the results presented in Chapter 4, we could also determine optimal settings for τ by performing a line search. Besides assuming an identity matrix for the observation model and restricting forcing to read-out states, we can more generally co-train some parameterized (possibly nonlinear) operator $\hat{\boldsymbol{z}}_t = B_\theta(\boldsymbol{x}_t)$ to infer control states from the data.

3.2.4 *Generalized Teacher Forcing (GTF)*

Generalized TF (GTF) [152] is similar in spirit to STF, but instead of fully (but sparsely in time) replacing latent states by model inferred states as in STF, the idea is to interpolate them during training with some constant $0 \leqslant \alpha \leqslant 1$, according to:

$$\tilde{\boldsymbol{z}}_t := (1-\alpha)\boldsymbol{z}_t + \alpha\hat{\boldsymbol{z}}_t, \tag{41}$$

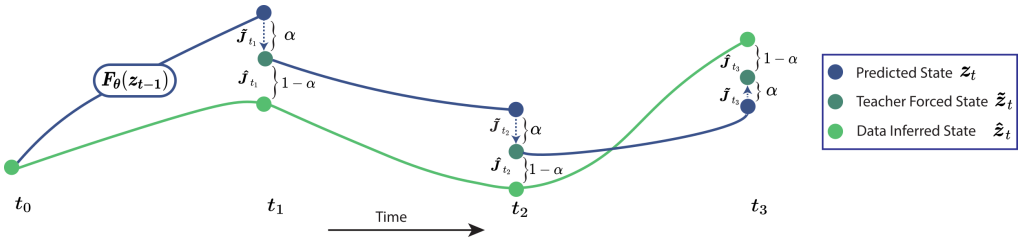

Figure 10: Principle of Generalized Teacher Forcing. Taken from [152].

Here, the RNN-predicted states $\boldsymbol{z}_t$ evolve according to Eq. 22, the data-inferred states are given as before by $\hat{\boldsymbol{z}}_t := \boldsymbol{B}^+\boldsymbol{x}_t$, where $\boldsymbol{B}^+$ is again the (pseudo-)inverse of $\boldsymbol{B}$, and the $\tilde{\boldsymbol{z}}_t$ are the teacher forced states given by a combination of forced and unforced states. Figure 10 illustrates the general procedure and notation. While the idea behind GTF can also already be traced back to the 1990s [88], a thorough theoretical and empirical study in the context of DSR was only carried out by us in [152].

As with STF, an appropriate choice of the hyperparameter α can fully remedy the exploding gradients for chaotic DS described in Sect. 3.2.1. This can be seen by considering how GTF affects the system's Jacobians. Using the chain rule

and plugging in Eq. 41, the Jacobians of the TF states scale proportionally to the strength of the TF signal $1-\alpha$:

$$\begin{aligned} \boldsymbol{J}_t &= \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_{t-1}} = \frac{\partial \boldsymbol{z}_t}{\partial \tilde{\boldsymbol{z}}_{t-1}} \frac{\partial \tilde{\boldsymbol{z}}_{t-1}}{\partial \boldsymbol{z}_{t-1}} = \frac{\partial \boldsymbol{F}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}_{t-1})}{\partial \tilde{\boldsymbol{z}}_{t-1}} \frac{\partial \tilde{\boldsymbol{z}}_{t-1}}{\partial \boldsymbol{z}_{t-1}} \\ &= (1-\alpha)\tilde{\boldsymbol{J}}_t, \end{aligned} \tag{42}$$

Derivatives of temporally distant states are then given by a product of Jacobians over forced states, multiplied with an exponential decay term scaling with the strength of α:

$$\begin{aligned} \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_r} = \prod_{k=0}^{t-r-1} \boldsymbol{J}_{t-k} &= \prod_{k=0}^{t-r-1} (1-\alpha)\tilde{\boldsymbol{J}}_{t-k} \\ &= (1-\alpha)^{t-r} \prod_{k=0}^{t-r-1} \tilde{\boldsymbol{J}}_{t-k}. \end{aligned} \tag{43}$$

This already provides some intuition into why GTF allows us to control the Jacobian product norm along trajectories. Particularly, considering all Jacobians of an RNN model $\mathcal{J} = \{\tilde{\boldsymbol{J}}_\kappa\}_{\kappa \in \mathcal{K}}$, we can define

$$\tilde{\sigma}_{max} := \sup \left\{ \left\| \tilde{\boldsymbol{J}}_\kappa \right\| = \sigma_{max}(\tilde{\boldsymbol{J}}_\kappa) \,:\, \tilde{\boldsymbol{J}}_\kappa \in \mathcal{J} \right\}, \tag{44}$$

For an RNN reconstructing a chaotic DS, $\tilde{\sigma}_{max} > 1$ [152], and hence the product in Eq. 43 diverges for T towards infinity. However, the decay term $(1-\alpha)^{t-r}$ can compensate for this divergence: ideally, it should balance the product so that gradient divergence is avoided without pushing gradients too far toward zero. Particularly, for most distant states in the product series, for which this divergence is expected to be most significant, this product should remain balanced around one:

$$\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} = (1-\alpha)^{T-1} \prod_{k=0}^{T-2} \tilde{\boldsymbol{J}}_{T-k} \overset{!}{=} \mathbb{1}. \tag{45}$$

These considerations imply that an optimal α can be chosen based on estimates of the system's Jacobians. In practice, different ways for using and estimating α during training exist, discussed in more detail in [152]. One straightforward way of directly relating α to σ_{max} is given by taking $\alpha := 1 - \frac{1}{\tilde{\sigma}_{max}}$. This guarantees that the Jacobian product series is bounded from above [152]. Using this formula requires obtaining a sensible estimate for $\tilde{\sigma}_{max}$. However, the theoretically most principled approach for estimating $\tilde{\sigma}_{max}$, based on Eq. 44, requires the evaluation of the full set $\mathcal{J}$ of the DSR model, which scales exponentially with model size for the PLRNN architectures discussed in the previous section (Sect. 3.1), since it entails evaluating the Jacobians in every linear subregion of the DSR model separately.

Since mimicking the dynamics on the training data is the goal of RNN training, a sensible and less computationally intense estimate $\tilde{\sigma}_{max}$ can be obtained by directly considering the Jacobians along training trajectories given by the TF states $\tilde{\boldsymbol{z}}_t$, and taking the maximum:

$$\hat{\sigma}_{max} = \max_t \left\| \tilde{\boldsymbol{J}}_t \right\|, \tag{46}$$

These estimates guarantee an α that formally avoids divergence by providing an upper bound on the Jacobian product series. In practice, however, they provide an overly conservative estimate for the TF signal, where the desired balance in Eq. 45 is not necessarily provided. A more practical way of estimating α is found by selecting α based on the condition in Eq. 45. Assuming non-singular Jacobians, this implies that

$$(1-\alpha)\mathcal{G}(\tilde{\boldsymbol{J}}_{T:2}) \stackrel{!}{=} \mathbb{1}, \tag{47}$$

$$\text{where} \quad \mathcal{G}(\tilde{\boldsymbol{J}}_{T:2}) := \left(\prod_{k=0}^{T-2} \tilde{\boldsymbol{J}}_{T-k}\right)^{\frac{1}{T-1}}. \tag{48}$$

Assuming we estimate the Jacobians $\hat{\boldsymbol{J}}_t$ from a training sequence, evaluated at data-inferred states, this yields:

$$\alpha = \left[1 - \frac{1}{\left\|\mathcal{G}(\hat{\boldsymbol{J}}_{T:2})\right\|}\right]. \tag{49}$$

This estimate can also be taken over several p training batches, using $\alpha = \max_p \alpha^{(p)}$. While computing this estimate again necessitates evaluating products of Jacobians, whose divergence in the context of chaotic system motivates using TF techniques in the first place, approximations for these products of Jacobians can be found to obtain estimates for α during training (details are given in [152]). Lastly, α can also be treated adaptively during the training process, e.g. by starting with a strong forcing signal when observations do not yet match the true dynamics, and then slowly phasing out forcing as the model increasingly learns to approximate system dynamics (see also Fig. 11). This idea can also be combined with iteratively updating estimates of α from Eq. 48.

However, as for STF, α can also be treated as a hyperparameter optimized by line search. For several benchmark systems, reconstructions were also not overly sensitive to the precise choice of α, achieving good reconstructions along a range of values [152], and α optimized via line search performed similarly to the adaptive α scheme (shPLRNN+GTF and shPLRNN+aGTF in Table 7). An important observation is that GTF smoothens loss landscapes (see also Figs. 13 and 14 illustrating the same effect for MTF), which has previously been noted in the context of other control theoretic approaches [3]. This can intuitively be understood by noting that small changes in parameters lead to less dramatic changes in overall dynamics: just as the product of Jacobian in Eq. 43 is smoothened out by an exponential decay term weighted by $1-\alpha$, differences in dynamics caused by small changes in parameter do not lead to equally exponential changes in predicted trajectories that would naturally occur in chaotic DS, and hence to less abrupt changes in the resulting losses. However, as illustrated in Figs. 13 and 14, overly strong TF signals lead to over smoothed loss landscapes that even become insensitive to divergences in the dynamics of the DSR model. Finally, Fig. 11 summarizes example training sequences at different stages of training using STF and GTF.

3.2.5 *Multimodal Teacher Forcing (MTF)*

The multimodal TF (MTF) framework [53] combines many of the ideas discussed in the previous sections. While STF (Sect. 3.2.3) and GTF (Sect. 3.2.4) are

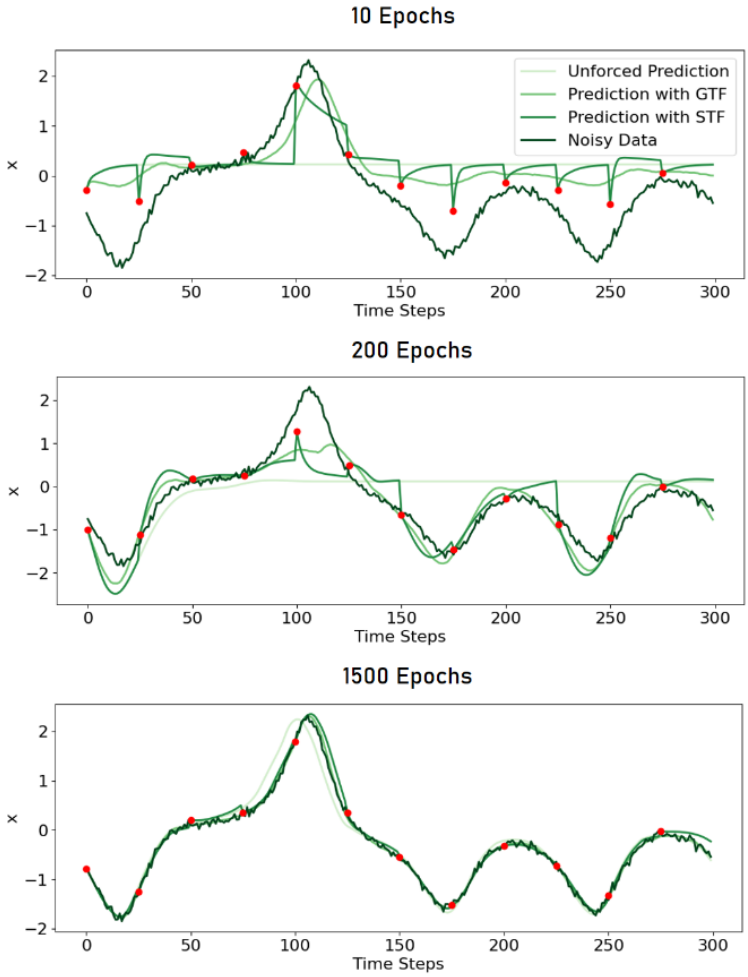

Figure 11: Example training sequences ($T_{seq} = 300$) at different stages of training using STF ($\tau = 25$, with forcing times highlighted in red), and GTF ($\alpha = 0.15$) for training a shPLRNN on the Lorenz-63 system. Forced states with STF do not align perfectly with the data since forcing occurs prior to the RNN step. Note that training on this sequence length without any TF quickly leads to divergences. Hence the unforced prediction (light green) is drawn from the model trained with GTF, and intended to serve as a reference for the freely evolving model predictions at this epoch.

effective at obtaining balanced gradients during training, thereby allowing training on long sequences, they embed the reconstruction model in a deterministic framework and rely on the inversion of the observation model to infer TF states. Inverting the observation model, however, is not always possible, since so far we have assumed a relatively straightforward relationship between the latent states of the DSR model and observations via the linear Gaussian observation model (Eq. 14) and the identity observation model (Eq. 37). However, relating back to the dual challenge inherent in data-driven DSR discussed in the introduction (Sect. 2.3, [218]), it is often a crucial component of a DSR algorithm to discover an appropriate coordinate system within which to represent the reconstructed latent DS, especially if the relationship between measurements and the latent DS is increasingly complicated, such as for partial observations, discrete random variables or combinations of jointly observed time series.

The SVAE (Sect. 3.2.2) trains the DSR model within a flexible encoder/decoder architecture (Fig. 9). Kramer et al. [214] integrate the SVAE in a multi-

modal setting, combining multiple different continuous and/or discrete data channels into the same reconstruction model. However, as discussed in Sect. 3.2.2 (see also Fig. 13), SVAEs perform much worse on experimental data and challenging DS benchmarks than approaches based on TF, as will also be confirmed by the benchmark comparisons presented in Sect. 4.3.1.

Inspired by the respective strengths of SVAEs and TF-based training, MTF provides a novel and comprehensive framework for multimodal data integration for DSR. The central idea is to use a Multimodal Variational Autoencoder (MVAE) to create a joint latent representation across different observed data channels and potentially different data types (continuous and/or discrete). This latent representation then provides a (sparse) TF signal during the training of a DSR model. Both the DSR model and the MVAE are then coupled to the observations through a set of shared, modality-specific decoder (observation) models that take the distinct statistical properties of the observations into account. This approach is illustrated in Fig. 12.

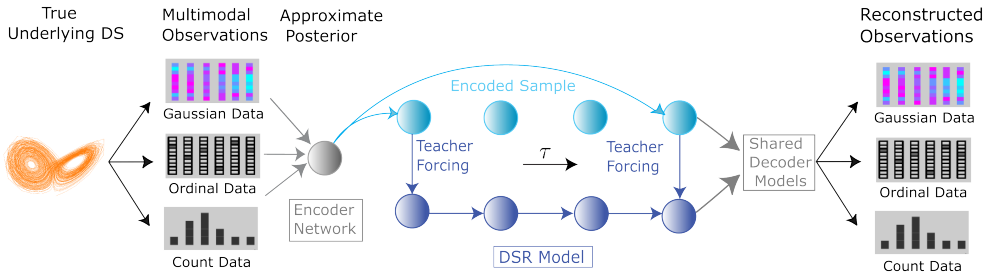

Figure 12: Principle of Multimodal Teacher Forcing. Taken from [53].

This exposition follows the one in [53] with some adjustments in notation to ensure consistency with the rest of this chapter. As in [53], here I assume training with STF, but the ideas behind GTF are also naturally accommodated into MTF by incorporating the MTF states in Eq. 41.

Consider a combination of multivariate Gaussian ($\boldsymbol{X}$), ordinal ($\boldsymbol{O}$), and count ($\boldsymbol{C}$) observations of length T, constituting the training data

$$\boldsymbol{Y} = \{\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}; \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}; \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_T\}\}.$$

We assume these are generated by a sequence of M-dimensional latent states of a general DSR model (Eq. 22) e.g. given by a dendPLRNN, $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T\}$, $\boldsymbol{z}_t \in \mathbb{R}^M$. In this case, we may take modality-specific decoder models such as the already introduced linear Gaussian model (see Eq. 50), a cumulative link model for ordinal data, and a log-link function for Poisson data (see Sect. 3.2.7 for details):

$$\boldsymbol{x}_t | \boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{B}\boldsymbol{z}_t, \boldsymbol{\Gamma}); \tag{50}$$

$$\boldsymbol{o}_t | \boldsymbol{z}_t \sim \text{Ordinal}(\boldsymbol{\beta}\boldsymbol{z}_t, \epsilon); \tag{51}$$

$$\boldsymbol{c}_t | \boldsymbol{z}_t \sim \text{Poisson}(\lambda(\boldsymbol{z}_t)). \tag{52}$$

Other combinations of modalities can naturally be accommodated (e.g. continuous neural recordings with categorical behavioral labels or discrete count spike trains combined with continuous position data, Sect. 4.3.3). The framework can also be used to train solely on discrete data, such as ordinal or count

data (Sect. 4.3.2). The loss of the DSR model is computed by summing the negative log-likelihoods of the respective observation models, which we here assume to be conditionally independent given the latent state $\boldsymbol{z}$:

$$\mathcal{L}_{\text{DSR}} = -\sum_{t=1}^{T} (\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_{1:K,t}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{o}_t|\boldsymbol{z}_{1:K,t}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{c}_t|\boldsymbol{z}_{1:K,t})), \quad (53)$$

The MVAE is used to encode a set of control states $\hat{\boldsymbol{Z}} = \{\hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_T\}$, $\hat{\boldsymbol{z}}_t \in \mathbb{R}^K$ jointly from the observations via the approximate posterior distribution $p_{\boldsymbol{\phi}}(\hat{\boldsymbol{Z}}|\boldsymbol{Y})$. This can, as for the SVAE in Sect. 3.2.2, be parameterized in different ways, e.g. via temporal convolutions (Sect. 3.2.6). The encoded states will serve as the (sparse) TF signals during the training of the DSR model. As for the SVAE, the MVAE is trained to minimize the negative Evidence Lower Bound (ELBO), here including all multimodal observations $\boldsymbol{Y}$:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}; \boldsymbol{Y}) = &- \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\boldsymbol{Y}|\hat{\boldsymbol{Z}}) \\ &+ \log p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Z}})] - \mathbb{H}_{q_{\boldsymbol{\phi}}}(\hat{\boldsymbol{Z}}|\boldsymbol{Y})\end{aligned} \quad (54)$$

with $\mathbb{H}_{q_{\boldsymbol{\phi}}}$ the entropy term as in Eq. 29. To ensure consistency between both latent codes, the control states are coupled to the same set of decoder models as the DSR model:

$$\boldsymbol{x}_t|\hat{\boldsymbol{z}}_t \sim \mathcal{N}(\boldsymbol{B}\hat{\boldsymbol{z}}_t, \boldsymbol{\Gamma}); \quad (55)$$

$$\boldsymbol{o}_t|\hat{\boldsymbol{z}}_t \sim \text{Ordinal}(\boldsymbol{\beta}\hat{\boldsymbol{z}}_t, \epsilon); \quad (56)$$

$$\boldsymbol{c}_t|\hat{\boldsymbol{z}}_t \sim \text{Poisson}(\lambda(\hat{\boldsymbol{z}}_t)), \quad (57)$$

The first term of the ELBO is then similar to the DSR loss (Eq. 54), but evaluated for the control states using the shared decoder models:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{Y}|\hat{\boldsymbol{Z}}) = -\sum_{t=1}^{T} (\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\hat{\boldsymbol{z}}_t) + \log p_{\boldsymbol{\theta}}(\boldsymbol{o}_t|\hat{\boldsymbol{z}}_t) + \log p_{\boldsymbol{\theta}}(\boldsymbol{c}_t|\hat{\boldsymbol{z}}_t)), \quad (58)$$

The DSR states $\boldsymbol{Z} \in \mathbb{R}^M$ and MVAE states $\hat{\boldsymbol{Z}} \in \mathbb{R}^K$ are not required to have the same dimensionality, $K \leqslant M$. As in id-TF, this separates the K readout states of the DSR model and the $M - K$ unforced states that do not contribute to the loss but can increase the expressivity of the DSR model.

With the decoder and encoder of the MVAE specified, what remains is the prior over control states $p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Z}})$. Given that in the optimal case after training, control states should agree with the states of the DSR model, it is natural to assume that this prior in turn is given by the DSR model:

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Z}})] \approx &\frac{1}{L}\sum_{l=1}^{L}\sum_{t=1}^{T} -\frac{1}{2}\big(\log|\boldsymbol{\Sigma}| \\ &+ (\hat{\boldsymbol{z}}_t^{(l)} - \boldsymbol{\mu}_t)^\top \boldsymbol{\Sigma}^{-1}(\hat{\boldsymbol{z}}_t^{(l)} - \boldsymbol{\mu}_t) \\ &+ \text{const.}\big),\end{aligned} \quad (59)$$

When using MTF with STF, the terms in Eq. 59 where $t \in \mathcal{F}$ are evaluated before forcing, as otherwise they would trivially evaluate to zero.

This expression is similar to the third row in the joint likelihood in Eq. 33, where the means $\boldsymbol{\mu}_t = \mathbb{E}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$ are given by the DSR model. However, the

crucial difference between MTF and SVAEs lies in the fact that here we obtain the means by generating longer trajectories of length T of RNN states $\boldsymbol{Z}$ from the DSR model while applying STF/GTF, and only use the control states for initialization and TF. This allows the DSR model to freely evolve longer trajectories and leverage the advantages of training with BPTT. This in turn puts less burden on the approximate posterior to fully capture the long-term structure (since here its primary role is in providing a control signal). In contrast, for the SVAE, we only consider means that are forward propagated one time step from the encoded state at the previous time step $\boldsymbol{\mu}_t = \mathbb{E}(\boldsymbol{z}_t|\hat{\boldsymbol{z}}_{t-1})$, ensuring short-term consistency. These observations, and their effect on the loss curves of the DSR models, are summarized in Fig. 13.

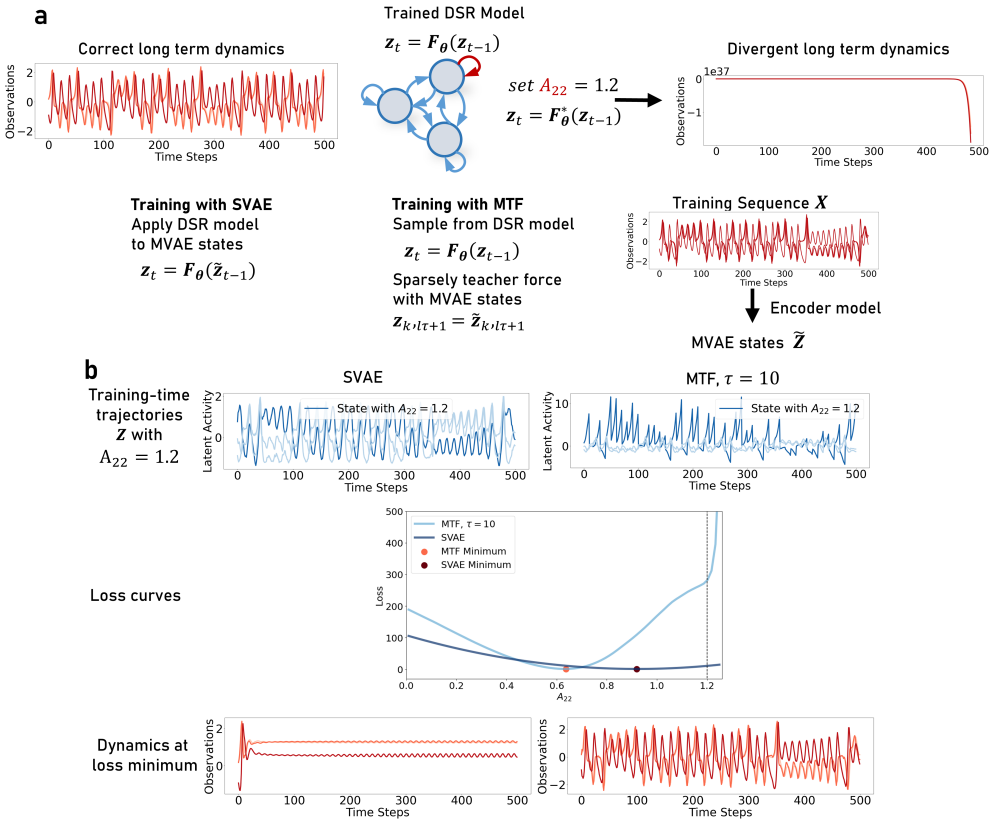

Figure 13: Illustration of the impact of dramatic changes in long-term dynamics on the SVAE and MTF loss. **a**: A dendPLRNN successfully trained on multimodal observations from the Lorenz-63 system is altered by setting a parameter of the linear self-connectivity $A_{22} > 1$, which results in globally diverging dynamics, while still looking locally consistent with the Lorenz-63. **b**: The global divergence is reflected in the training-time trajectories $\boldsymbol{Z}$ generated using MTF with interval $\tau = 10$ (right), within which the DSR model evolves freely. This divergence leads to large increases in the MTF training loss (see MTF loss curve for $A_{22} > 1$), and hence is strongly penalized. This effect is essentially not present for the SVAE, where the global divergence induces no considerable effect on the training loss and training-time trajectories. The mismatch in global (long-term) dynamics hence remains unrecognized by the SVAE. As shown at the bottom, at the minimum of the SVAE loss ($A_{22} \approx 0.966$) the dynamics converge to an equilibrium point (left), while MTF at its minimum ($A_{22} \approx 0.637$) produces trajectories which agree in their temporal structure with those of the original Lorenz-63 (right).

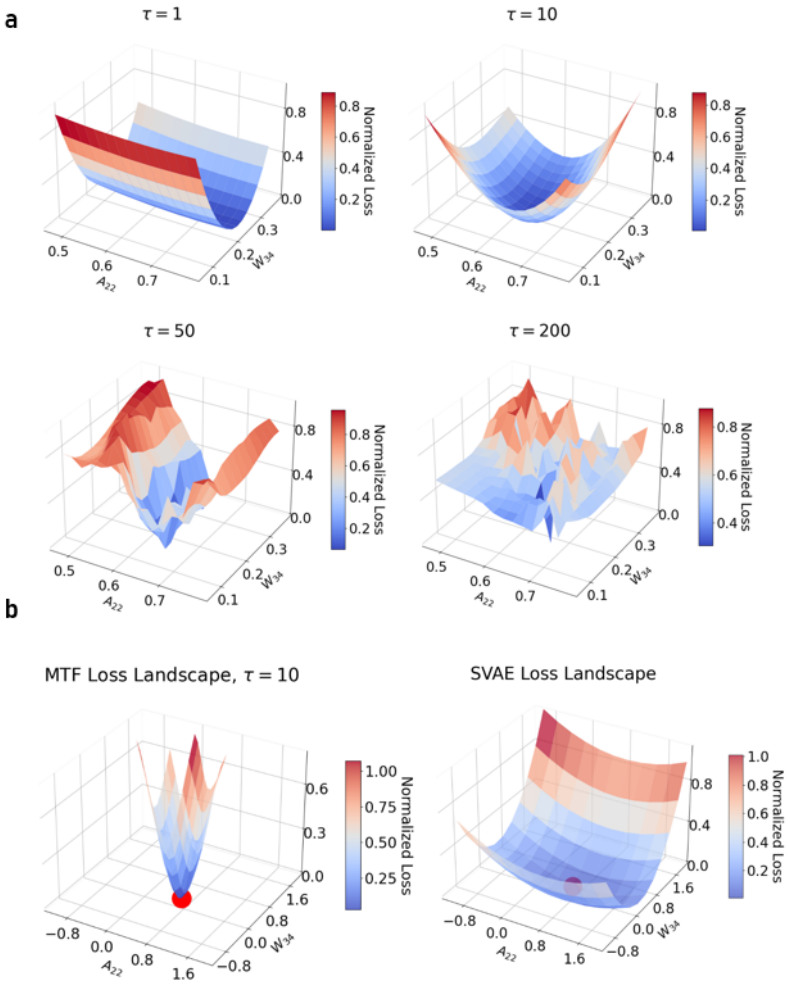

Figure 14: **a**: MTF loss landscapes, computed using the total loss (Eq. 60) by varying two parameters of a trained dendPLRNN (A_{22}, W_{34}) and computing the loss for a sequence of $T = 300$ time steps. Illustrated are four values of TF interval τ. Lower values of τ increasingly smoothen the loss landscape. $\tau = 10$ corresponds to an optimal choice for the TF interval, where the loss landscape appears both smoothed out and convex, while for low $\tau = 1$, the loss landscape flattens, making training more difficult. **b**: Comparison of MTF and SVAE loss landscapes. Since the SVAE loss (Eq. 33) only includes one-step ahead predictions from the DSR model, it essentially over-smoothens the loss landscape, similar to the observations made when choosing a very small τ in MTF, not allowing the model to evolve freely during training. Note that the parameter range that can be meaningfully explored for the MTF is smaller than for the SVAE since larger variations in the parameters (e.g. a value of A_{22} over 1) induce divergences in the sequences drawn from the DSR model for the computation of the DSR loss (see also Fig. 13). Based on [53].

Using the DSR model as a prior for the MVAE ensures consistency between both latent codes (top and bottom row in Fig. 12) and is therefore called *consistency loss*, $\mathcal{L}_{\text{con}}$. The total loss for the MTF framework is then given by the consistency loss, the DSR loss (Eq. 53, and the remaining ELBO terms (Eq. 54):

$$\mathcal{L}_{\text{MTF}} = \mathcal{L}_{\text{con}} + \mathcal{L}_{\text{DSR}} - \mathbb{E}_{q_\phi}[\log p_{\boldsymbol{\theta}}(\boldsymbol{Y}|\hat{\boldsymbol{Z}})] - \mathbb{H}_{q_\phi}(\hat{\boldsymbol{Z}}|\boldsymbol{Y}) \tag{60}$$

Fig. 15 demonstrates that all components of the loss contribute meaningfully to successful reconstructions. As already mentioned for GTF (see [152]) MTF smoothens the loss landscape, and an optimal TF interval makes the loss landscape convex and well-navigable around optimal parameter values (Fig. 14).

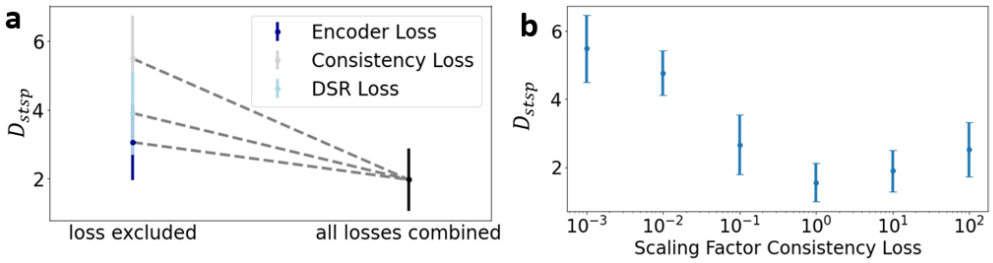

Figure 15: Comparison of state space agreement D_{stsp} in two scenarios (see Sect. 4.1 for details): (**a**) when omitting different terms from the total loss as specified in Eq. 60, and (**b**) while altering the scaling of the consistency loss $\mathcal{L}_{\text{con}}$. These comparisons are made for a dendPLRNN trained using MTF on multimodal Gaussian, ordinal, and count data from the chaotic Lorenz system (Fig. 24). Taken from [53].

3.2.6 *Encoder Models*

This section summarizes the encoder models used when training within the SVAE and MTF approach.

TEMPORAL CNN ENCODER The mean $\boldsymbol{\mu}_{t,\phi}$ of the CNN Encoder is modeled using stacked temporal convolutions, which process the inputs $\boldsymbol{x}_{t-w}...\boldsymbol{x}_{t+w}$, where the maximum kernel size determines w. In many real-world systems, such as chaotic systems, future values become decorrelated from the past after characteristic time-scales [127], and hence an optimal kernel size depends on the properties of the underlying DS [233]. In [54], we used a four-layer stack of temporal CNNs with progressively reducing kernel sizes (e.g. 41, 31, 21, and 11) for the mean, which are then mapped to the parameters of the approximate posterior. The diagonal covariance is directly mapped from the observations to the logarithms of the covariance using a single convolutional layer. The rationale for employing CNNs is based on the assumption that the data exhibits translationally invariant features in time, enabling the encoder model to integrate meaningful temporal context into its latent representation [78, 227]. This is exemplified by the observation that the CNN encoder effectively learns a temporal delay embedding when trained on partial observations of the Lorenz-63 attractor (Fig. 27). For image data, deep CNNs with small kernel sizes are the most popular choice [195, 397], while in the context of TSF and TS classification, wider kernels are often used [233]. For real-world applications where forecasting is the primary goal and inputs from the future are unrealistic (as for the trust game data in Sect. 5), the CNN encoder can implement this causal structure by only encoding past values $\boldsymbol{x}_{t-w}...\boldsymbol{x}_t$.

MIXTURE-OF-EXPERTS CNN ENCODER The Mixture-of-Experts (MoE) [366] and product-of-experts (PoE) [154, 426] are extensions of the temporal CNN encoder to multimodal settings that combine distinct encodings for each modality

into a combined estimate. For the modality-specific estimates, we used the same CNN encoder as above but trained a distinct encoder model for each modality individually. For the MoE, the outputs of each encoder are then combined into a joint estimate by either a weighted average :

$$\boldsymbol{\mu}_{\text{MoE}} = w_g \boldsymbol{\mu}_g + w_o \boldsymbol{\mu}_o + w_c \boldsymbol{\mu}_c$$
$$\boldsymbol{\Sigma}_{\text{MoE}} = w_g \boldsymbol{\Sigma}_g + w_o \boldsymbol{\Sigma}_o + w_c \boldsymbol{\Sigma}_c,$$

with means $\boldsymbol{\mu}_{\{g,o,c\}}$, diagonal covariances $\boldsymbol{\Sigma}_{\{g,o,c\}}$ of the respective experts, and mixing weights w_g (Gaussian), w_o (ordinal) and w_c (count process), which can either be co-trained or set to constant values, here chosen to be $1/3$ for each modality. The MoE can for instance be beneficial for cross-modal inference in the case where individual time series contain missing entries (see Fig. 26 for DSR from both Gaussian and ordinal observations in a situation where 20% of the time steps are randomly missing in each modality).

For the PoE, the estimates of the individual experts are multiplied instead of summed, which however often led to numerical instabilities during training from discrete variables.

RNN ENCODER We further tested an RNN encoder [73], where the hidden states $\boldsymbol{h}_t$ of an RNN are mapped onto the parameters of the approximate posterior at every time step. Following the standard RNN model in `torch.nn.RNN`, this results in:

$$\boldsymbol{h}_t = \tanh(\boldsymbol{W}\boldsymbol{h}_{t-1} + \boldsymbol{U}\boldsymbol{y}_t + \boldsymbol{b})$$
$$\boldsymbol{\mu}_t = \boldsymbol{W}_\mu \boldsymbol{h}_t + \boldsymbol{b}_\mu$$
$$\mathrm{diag}\left([\log \sigma_1^2, \ldots, \log \sigma_K^2]\right) = \boldsymbol{W}_\Sigma \boldsymbol{h}_t + \boldsymbol{b}_\Sigma$$

with trainable RNN parameters $\{\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{b}\}$ and linear readout weights $\{\boldsymbol{W}_\mu, \boldsymbol{b}_\mu\}$ for the mean and $\{\boldsymbol{W}_\Sigma, \boldsymbol{b}_\Sigma\}$ for the logarithm of the diagonal covariance of the approximate posterior, respectively. The observations $\boldsymbol{y}_t$ are hence provided as input at every time step and the model is trained non-autonomously, corresponding to the 'classical' RNN training setup (see Sect. A.2). Therefore, the model will usually reflect the sub-par performance of classical RNN training on DSR tasks (see for instance comparison to 'BPTT' in Table 5). If instead the RNN encoder is evolved freely for longer times, which would be beneficial in a DSR context, it either requires a TF signal, or longer sequence lengths will inevitably lead to the same gradient problems described in Sect. 3.2.1. This somewhat tautological issue of requiring to effectively train an RNN (the encoder model), which is then used to provide a TF signal to train another RNN (the DSR model), might help explain why this formulation did not achieve comparable performance (Table 2).

TRANSFORMER We also implemented a Transformer encoder based on [406]. Given the time series nature of the data, we used positional encodings as proposed by Vaswani et al. [406]. The input time series, augmented with embeddings, was processed through a typical Transformer encoder block, using the default implementation in Pytorch `torch.nn.TransformerEncoder`. The output, as for the other encoder models, was then mapped to the mean and logarithm of the covariance of the approximate posterior through linear readout layers.

MULTI-LAYER PERCEPTRON (MLP) Lastly, we also implemented an MLP encoder, comprising 3 fully connected layers with ReLU nonlinearity. Again, the MLP output is mapped to the mean and logarithm of the covariance of the approximate posterior via a linear readout layer.

Table 2: Performance comparison of encoder and RNN models trained using MTF on multimodal data from the chaotic Lorenz system, using the performance metrics introduced in Sect. 4.1. Taken from [53].

Encoder/RNN Model	$D_{stsp}\downarrow$	$D_H\downarrow$	PE $\downarrow$	OPE $\downarrow$	SCC $\downarrow$	OACF $\downarrow$	CACF $\downarrow$
CNN	**3.4 ± 0.35**	**0.30 ± 0.06**	**1.3e−2 ± 2e−4**	**0.12 ± 0.03**	**0.07 ± 0.01**	**0.07 ± 0.01**	**6.6e−5 ± 8e−6**
CNN-MoE	5.89 ± 0.18	0.43 ± 0.03	2.3e−2 ± 5e−4	0.13 ± 0.00	0.10 ± 0.00	0.19 ± 0.01	1.1e−4 ± 2e−5
RNN	5.47 ± 0.48	0.32 ± 0.04	1.6e−2 ± 2e−4	0.15 ± 0.01	0.13 ± 0.02	0.05 ± 0.01	8.5e−5 ± 9e−6
Transformer	5.85 ± 0.14	0.40 ± 0.04	4.8e−2 ± 5e−4	0.16 ± 0.00	0.17 ± 0.03	0.16 ± 0.02	9.5e−5 ± 7e−6
MLP	6.57 ± 0.14	0.43 ± 0.01	5.4e−2 ± 6e−4	0.15 ± 0.00	0.15 ± 0.01	0.21 ± 0.01	1.3e−4 ± 9e−6
dendPLRNN	**3.4 ± 0.35**	**0.30 ± 0.06**	**1.3e−2 ± 2e−4**	**0.12 ± 0.03**	**0.07 ± 0.01**	**0.07 ± 0.01**	**6.6e−5 ± 8e−6**
LSTM	3.8 ± 0.74	0.31 ± 0.01	5.4e−2 ± 5e−4	0.16 ± 0.03	0.09 ± 0.02	0.09 ± 0.02	8.8e−5 ± 8e−6
GRU	**3.47 ± 0.56**	**0.29 ± 0.03**	3.5e−2 ± 5e−4	0.13 ± 0.03	**0.06 ± 0.01**	**0.08 ± 0.01**	7.1e−5 ± 5e−6

3.2.7 *Decoder Models*

ORDINAL DECODER MODEL For ordinal data, there is a natural ordering between variables, e.g. in survey data in economy or psychology commonly assessed through Likert scales [241], often ranking from 1 to 7. Treating ordinal data as metric can lead to a variety of problems, as pointed out in [240]. Here ordinal observations are coupled to latent states via a generalized linear model [267]. Specifically, we assume that the ordinal observations $\boldsymbol{o}_t$ relate to an underlying continuous variable u_{it}, which is linked to latent states $\boldsymbol{z}_t$ via a linear model

$$u_{it} = \boldsymbol{\beta}_i^T \boldsymbol{z}_t + \epsilon_{it}, \tag{61}$$

where $\boldsymbol{\beta}_i^T \in \mathbb{R}^M$ are the model parameters and ϵ_{it} is an independently distributed noise term. The distributional assumptions about the noise term ϵ_{it} determine which link function to use. A Gaussian assumption leads to an ordered probit model, while a logistic assumption leads to an ordered logit model [423]. While both models lead to relatively similar results in preliminary experiments, we found the ordered logit model to work slightly better in practice for the results in [53]. Inverting the link function leads to an expression for the cumulative probabilities:

$$p\left(o_{it} \leqslant k | \boldsymbol{z}_t\right) = \frac{\exp\left(\beta_{ik}^0 - \boldsymbol{\beta}_i^T \boldsymbol{z}_t\right)}{1 + \exp\left(\beta_{ik}^0 - \boldsymbol{\beta}_i^T \boldsymbol{z}_t\right)}. \tag{62}$$

The probabilities $p\left(o_{it} = k | \boldsymbol{z}_t\right)$ follow from the cumulative distribution by subtracting neighboring cumulative probabilities $p\left(o_{it} = k | \boldsymbol{z}_t\right) = p\left(o_{it} \leqslant k | \boldsymbol{z}_t\right) - p\left(o_{it} \leqslant k-1 | \boldsymbol{z}_t\right)$, which finally gives the log-likelihood as

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{O}|\boldsymbol{Z}) = \sum_{i}^{N} \sum_{t}^{T} \sum_{k}^{K} [o_{it} = k] \log p\left(o_{it} = k | \boldsymbol{z}_t\right). \tag{63}$$

CATEGORICAL MODEL Categorical observations are, like ordinal observations, not associated with a metric space, but in contrast to ordinal data, there

is also no natural ordering between the variables. To couple categorical observations to the latent states, we employed the natural link function given by

$$\pi_i = \frac{\exp\left(\boldsymbol{\beta}_i^T \boldsymbol{z}_t\right)}{1 + \sum_{j=1}^{K-1} \exp\left(\boldsymbol{\beta}_j^T \boldsymbol{z}_t\right)} \quad \forall i \in \{1 \dots K-1\} \tag{64}$$

$$\pi_K = \frac{1}{1 + \sum_{j=1}^{K-1} \exp\left(\boldsymbol{\beta}_j^T \boldsymbol{z}_t\right)}$$

Here, the parameters $\boldsymbol{\beta}_i \in \mathbb{R}^{M \times 1}$ constitute the respective regression weights for category $i = 1 \dots K-1$, with the total probability over all categories $\sum_{i=1}^{K} \pi_i = 1$. This leads to the following log-likelihood for the categories:

$$\log p_\theta(\boldsymbol{O}|\boldsymbol{Z}) = \sum_{i}^{N} \sum_{t}^{T} \sum_{k}^{K} [o_{it} = k] \log \left(\frac{\exp\left(\boldsymbol{\beta}_k^T z t\right)}{1 + \sum_{j=1}^{K-1} \exp\left(\boldsymbol{\beta}_j^T \boldsymbol{z}_t\right)} \right)$$

where $[o_{it} = k]$ is an indicator function that is 1 if the observation o_{it} belongs to category k and 0 otherwise.

POISSON MODEL For count observations $\{\boldsymbol{c}_t\}_{t=1}^{T}$, with $\boldsymbol{c}_t = (c_{1t}, \dots, c_{Lt})^T$, we tested three different decoder models. First, a standard Poisson model, with probabilities

$$p_\theta(c_{lt}|\boldsymbol{z}_t) = \frac{\lambda_{lt}^{c_{lt}}}{c_{lt}!} e^{-\lambda_{lt}}. \tag{65}$$

The probabilities are related to the latent states via the log-link function, $\log \lambda_{lt} = \gamma_0^{(l)} + \sum_{m=1}^{M} \gamma_m^{(l)} z_{mt}$, where $\boldsymbol{\gamma}^{(l)}$ is a vector of coefficients. Thus, $\lambda_{lt} = e^{\gamma_0^{(l)} + \boldsymbol{\gamma}^{(l)} \boldsymbol{z}_t}$ is the expected count for the l^{th} observation at time t. The total log-likelihood for observed counts $\boldsymbol{C}$ is then given by:

$$\log p_\theta(\boldsymbol{C}|\boldsymbol{Z}) = \sum_{t=1}^{T} \sum_{l=1}^{L} \left[c_{lt} \log \lambda_{lt} - \lambda_{lt} - \log(c_{lt}!) \right], \tag{66}$$

ZERO-INFLATED POISSON MODEL Alternatively, we tested a zero-inflated Poisson (ZIP) observation model [223]. The ZIP model is designed for count data that has an excess of zero counts compared to what would be expected from a traditional Poisson distribution, and thus can naturally handle overdispersion often observed in real data, such as the spike trains displayed in Fig. 32. The ZIP model addresses this by combining a binary process that determines whether a count is zero with probability π_{lt}) or follows a Poisson distribution. The combined probability of an observed count given both processes is then given by

$$p_\theta(c_{lt} \mid \boldsymbol{z}_t) = \begin{cases} \pi_{lt} + (1 - \pi_{lt})\, e^{-\lambda_{lt}} & \text{if } c_{lt} = 0 \\ (1 - \pi_{lt})\, \frac{\lambda_{lt}^{c_{lt}}}{c_{lt}!} e^{-\lambda_{lt}} & \text{if } c_{lt} > 0 \end{cases} \tag{67}$$

The probabilities are connected to the latent states via a logit and a log-link function $\log \frac{\pi_{lt}}{1-\pi_{lt}} = \beta_0^{(l)} + \sum_{m=1}^{M} \beta_m^{(l)} z_{mt}$ and $\log \lambda_{lt} = \gamma_0^{(l)} + \sum_{m=1}^{M} \gamma_m^{(l)} z_{mt}$, where $\boldsymbol{\gamma}^{(l)}$ and $\beta^{(l)}$are coefficient vectors. Thus, $\lambda_{lt} = e^{\boldsymbol{\gamma}^{(l)} \boldsymbol{z}_t}$ is the expected count for the $l-$th observation and $\pi_{lt} = \frac{e^{\beta^{(l)} z_t}}{1 + e^{\beta^{(l)} z_t}}$ is the probability of observing a zero.

NEGATIVE BINOMIAL MODEL Finally, the count observations can be modeled by a negative binomial model, given by

$$p_\theta(c_{lt}|z_t) = \frac{\Gamma(c_{lt}+\phi_l)}{\Gamma(\phi_l)c_{lt}!}\left(\frac{\phi_l}{\mu_{lt}+\phi_l}\right)^{\phi_l}\left(\frac{\mu_{lt}}{\mu_{lt}+\phi_l}\right)^{c_{lt}}, \tag{68}$$

where μ_{lt} is the mean count and ϕ_l the dispersion parameter of the negative binomial distribution for the l^{th} observation at time t. As for the Poisson model, we used a log-link function, $\log \mu_{lt} = \gamma_0^{(l)} + \sum_{m=1}^{M} \gamma_m^{(l)} z_{mt}$, with $\gamma^{(l)}$ a vector of coefficients, and z_{mt} the m^{th} latent variable at time t. Properly accounting for dispersion significantly improved the modeling of the spike counts in Fig. 32.

3.3 HIERARCHIZATION FRAMEWORK

The DSR models and learning algorithms proposed in the previous sections can be naturally accommodated into a hierarchization framework. Previous work in our group by Abaigar [1] approached hierarchization by splitting the parameters of the dendPLRNN (Sect. 3.1.1) into group-level parameters, shared by all models, and subject-specific parameters of the dendritic basis expansion:

$$\begin{aligned} \theta_{\text{group}} &= \{A, W, h, \Sigma\} \\ \theta_{\text{subj}}^{(j)} &= \{\theta_{\text{obs}}^{(j)}, \theta_a^{(j)}, \theta_b^{(j)}, h^{(j)}\} \quad \text{for } j = 1, \dots, N_{\text{subj}} \end{aligned} \tag{69}$$

The idea is then to train the model simultaneously on data from all participants. During training, random batches $\boldsymbol{x}_{0\dots T}^{(j)}$ belonging to subject j are drawn from all observations $\boldsymbol{X}^N$ across all N subjects. The respective loss, given this minibatch $\boldsymbol{x}_{0\dots T}^{(j)}$, is then used to compute error gradients for both $\boldsymbol{\theta}_{\text{group}}$ and $\boldsymbol{\theta}_{\text{subj}}^{(j)}$. Depending on the training setup, gradient updates can be computed after every minibatch, or updates can be averaged across several minibatches. In the experiments in Sect. 4.4, I computed parameter updates after 16 minibatches, the same as for the results without hierarchization.

3.3.1 *Hierarchical shPLRNN*

While we assume that subjects share common statistical features with the group, the level of variation across subjects might differ across different studies, or subjects might fall into several clusters with relatively distinct properties. This implies that a hierarchization framework should be flexible enough to accommodate different levels of in-group variation in its formulation. Further, in the approach along the lines of Eq. 69, it is unclear how to optimally choose which parameters belong to $\boldsymbol{\theta}_{\text{group}}$ and which to $\boldsymbol{\theta}_{\text{subj}}^{(j)}$, and how this grouping affects expressivity of the framework.

These shortcomings motivated the development of a more flexible formulation for hierarchization in DSR models. In this approach, the main idea is to split the DSR model into a set of trainable weight vectors constituting subject-level parameters/features $\theta_{\text{subj}} = \boldsymbol{l}^{(j)}$ with N_{feat} free parameters, which during training are projected onto the parameters of the DSR model via projection matrices θ_{group} which are jointly trained and shared across subjects. Only the weight vectors $\boldsymbol{l}^{(j)}$ are then fine-tuned for each subject, reducing the inter-subject differences to an N_{feat}-dimensional parameter manifold. This approach is illustrated

in Fig. 16. Similar ideas of splitting model parameters into larger projection matrices and fine-tuning low-rank parameter vectors on individual tasks have been explored in different ML applications, for instance in generative modeling tasks using RNNs [375, 386] or restricted Boltzmann machines [391], or in the context of transfer learning and fine-tuning of large scale models on new tasks (such as Low-Rank Adaptation (LoRA) and its many variants [160] used on LLMs).

I will introduce this approach more formally for the example of a hierarchically trained shPLRNN (Eq. 18), but the method naturally extends to other DSR models by replacing the respective parameters. As previously noted, the subject level parameters θ_{subj} are captured by the feature vector $\boldsymbol{l}^{(j)} \in \mathbb{R}^{1\times N_{\text{feat}}}$, which we here take to be a row vector. This vector is used to generate the full parameter set of the DSR model through learned projection matrices:

$$\begin{aligned}
\boldsymbol{W}_1^{(j)} &:= \text{mat}(\boldsymbol{l}^{(j)} \cdot \boldsymbol{P}_{W_1}, M, L),\\
\boldsymbol{W}_2^{(j)} &:= \text{mat}(\boldsymbol{l}^{(j)} \cdot \boldsymbol{P}_{W_2}, L, M),\\
\boldsymbol{h}_1^{(j)} &:= \boldsymbol{l}^{(j)} \cdot \boldsymbol{P}_{h_1},\\
\boldsymbol{h}_2^{(j)} &:= \boldsymbol{l}^{(j)} \cdot \boldsymbol{P}_{h_2},\\
\boldsymbol{A}^{(j)} &:= \text{diag}(\boldsymbol{l}^{(j)} \cdot \boldsymbol{P}_A),
\end{aligned}$$

where $\text{mat}(\cdot, m, n)$ denotes the operation of reshaping a vector into an $m \times n$ matrix, $\boldsymbol{W}_1$ is the resulting $M \times L$ connectivity matrix, $\boldsymbol{W}_2$ is the resulting $L \times M$ connectivity matrix, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$ are the threshold vectors of dimensions M and L, and $\boldsymbol{P}_{W_1} \in \mathbb{R}^{N_{\text{feat}}\times(M\cdot L)}$, $\boldsymbol{P}_{W_2} \in \mathbb{R}^{N_{\text{feat}}\times(L\cdot M)}$, $\boldsymbol{P}_A \in \mathbb{R}^{N_{\text{feat}}\times M}$, $\boldsymbol{P}_{h_1} \in \mathbb{R}^{N_{\text{feat}}\times M}$, and $\boldsymbol{P}_{h_2} \in \mathbb{R}^{N_{\text{feat}}\times L}$ are the projection matrices for each respective parameter.

From this, we obtain the latent equation of a reparameterized shPLRNN, given a subject-specific parameter vector $\boldsymbol{l}^{(j)} \in \mathbb{R}^{N_{\text{feat}}}$, as:

$$\boldsymbol{z}_t = \boldsymbol{A}^{(j)}\boldsymbol{z}_{t-1} + \boldsymbol{W}_1^{(j)}\phi(\boldsymbol{W}_2^{(j)}\boldsymbol{z}_{t-1} + \boldsymbol{h}_2^{(j)}) + \boldsymbol{h}_1^{(j)}, \tag{70}$$

A crucial ingredient for successful training within this framework is to assign a considerably higher learning rate to the feature vector (around an order of magnitude) compared to the projection matrices. This helps ensure that a higher burden is put on the model to incorporate subject-specific information through the feature vector. It further avoids numerical instabilities which often occur when choosing higher learning rates for the projections. Another important aspect of successful training involves careful initialization of the projection matrices. Here I used a Xavier Uniform Initialization [130], which is designed to keep the variance of the outputs n_{out} of a layer roughly the same as the variance of its inputs n_{in} by drawing weights from a uniform distribution in the interval $[a, a]$, where $a = \sqrt{\frac{6}{n_{in}+n_{out}}}$. After drawing the weights I scaled down the initialized projection matrices by a factor of 0.1 to further reduce the variance of the outputs in the early stages of training. Again, this helps stabilize the learning process since in this formulation, slight changes in weights can lead to bifurcations in dynamics often observed in DSR [104, 149].

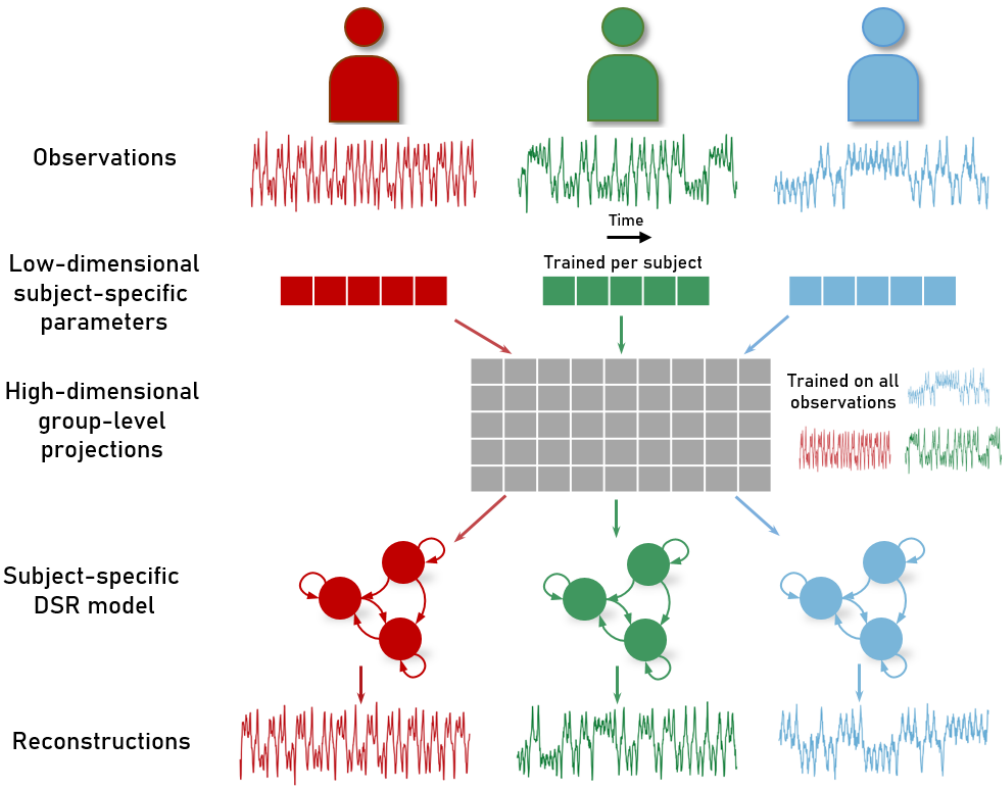

Figure 16: Illustration of the hierarchical inference framework.

Part III

RESULTS

4

RESULTS

This chapter recapitulates the main experimental results of [54], [152], and [53], some theoretical results from [139], a summary of the main results from [149], and unpublished results for the hierarchical inference framework and the linear regions analysis pipeline. These results include:

- The introduction of a range of performance metrics tailored to DSR for unimodal and multimodal data.
- The introduction of a statistical and topological error sensitive to multistability, and a discussion of OOD learnability in DSR.
- Extensive evaluations of the dendPLRNN and shPLRNN trained with STF, GTF and SVAE on several low-and high-dimensional unimodal benchmark systems.
- DSR from challenging experimental datasets.
- Comparisons with many other DSR algorithms, based e.g. on Reservoir Computers (RCs, [303]), Long-Short-Term Memory Networks (LSTMs, [408], Sparse Identification of Nonlinear Dynamical Systems (SINDy, [59]), Neural ODEs (Neural ODE, [70]) and Long-Expressive-Memory (LEM, [341].)
- Evaluations of MTF on a range of multimodal benchmarks, including comparisons with five other approaches for DSR from multimodal data, based on a sequential MVAE as proposed in [214], 'classical' RNN training with truncated backpropagation through time (BPTT), multiple shooting (MS), and two approaches with 'Gaussianized' data.
- DSR of chaotic systems purely from discrete ordinal and symbolic/categorical encodings, using MTF.
- DSR from two real-world multimodal datasets, using MTF.
- The extraction of low-dimensional parameter vectors from benchmark systems and experimental data using the hierarchization approach.
- The extraction of sparse interpretable graph structure from the linear subregions visited by PLRNNs trained on several benchmark systems.
- A novel pruning procedure based on geometric attractor agreement that leads to specific network topologies beneficial for DSR.

4.1 PERFORMANCE METRICS IN DSR

The training algorithms introduced in the previous chapter all rely on loss functions that incorporate some variant of a prediction error between model-generated predictions and observed data in the optimization criterion (e.g. the

MSE in Eq. 39, and the corresponding Gaussian likelihoods in Eq. 33). However, in DSR, we are not primarily interested in good short-term predictions, but rather in reproducing the long-term temporal, geometric and topological properties of the underlying system. Further, the MSE (and short-term prediction errors) is unsuitable for assessing reconstruction in chaotic DS for reasons touched upon before [139]: it is inadequate as a test loss due to exponential trajectory divergence in chaotic DS (cf. Eq. 24, Fig. 17**a** left), and does not necessarily capture long-term, invariant, or topological properties of DS. Particularly, it is not designed to be sensitive to multistability [139]. These shortcomings necessitate the introduction of new performance metrics that take the geometric, temporal, and topological structure of the reconstructed systems into account, ensuring that these are captured correctly.

4.1.1 *Performance Metrics for Unimodal Continuous Time Series*

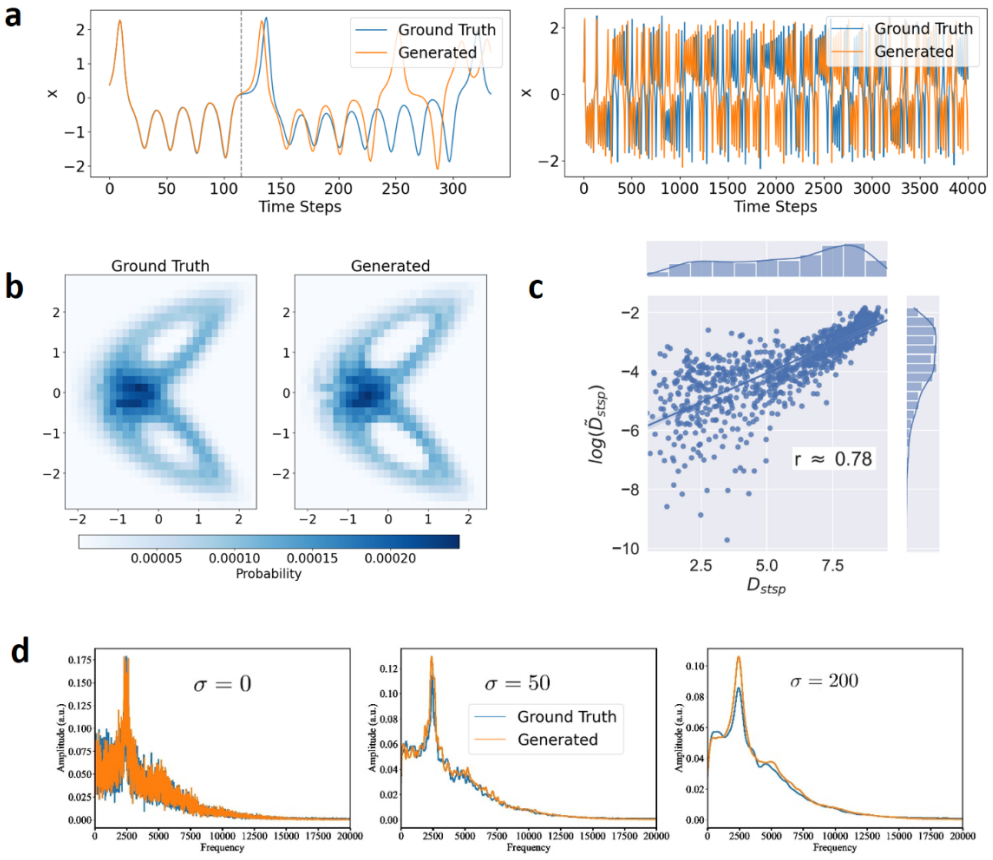

Figure 17: Overview over metrics for continuous data for models trained on the chaotic Lorenz-63 system. **a**: While short-term predictions deteriorate after a certain time horizon for chaotic systems due to exponential trajectory divergence (here after around 120 time steps), the long-term temporal patterns can still agree (right). **b**: State-space agreement approximated via the binning approximation, projected along the z-direction. **c**: Correlation between D_{stsp} approximated via the binning method ($m = 30$) and the logarithm of D_{stsp} approximated as a GMM for generated data from different trained models. **d**: Example power spectra for different smoothing factors σ. (**c**) and (**d**) taken from [54].

SHORT TERM PREDICTION ERROR The n-step ahead prediction error (PE) is defined as the mean squared error (MSE) between predicted and true observations:

$$\mathrm{PE}(n) = \frac{1}{N(T-n)} \sum_{t=1}^{T-n} \sum_{i=1}^{N} (x_{i,t+n} - \hat{x}_{i,t+n})^2. \tag{71}$$

The PE is computed by initializing the model with the test set time series up to a certain time point t, and then iterating it forward by n time steps to yield a prediction at time step $t+n$. Due to exponential trajectory divergence (Fig. 17**a**), the precise initialization at time t can significantly affect prediction performance. This has some practical implications since for some DSR algorithms initializing the reconstruction model is difficult, or not possible at all. In approaches where an encoder model is explicitly co-trained, such as the SVAE, the initial state $\boldsymbol{z}_t$ can be estimated from the data $p(\boldsymbol{z}_t|\boldsymbol{X})$. In applications where short-term forecasts are of primary interest (see Sect. 5), this often requires masking future values (causal masking), as otherwise future observations are taken into account in the estimation of the present state. This is often either not feasible in real-time applications or provides an unfair advantage in benchmark comparisons.

For training with STF, we estimated the initial state $\mathbf{z}_t$ by a co-trained linear mapping to $\mathbf{x}_t$ [54]. For MTF, the initial state $(\mathbf{z}_t|\boldsymbol{X})$ is directly obtained from the encoder model if $K = M$. In cases where $K < M$, the subset of $M - K$ unforced latent states needs to be initialized differently, e.g. by sampling from a normal distribution. In this under-specified scenario, I employed a warm-up phase of $t_w = 20$ time steps. This phase involved initializing the system from the encoder at time $t - t_w$ and iteratively providing encoded states as TF signals, resulting in an initial state estimate $\mathbf{E}[\mathbf{z}_t|\mathbf{x}_{(t-t_w):t}]$. Incorporating this warm-up phase significantly improved predictions, though it still yielded slightly inferior results compared to the fully specified case where $K = M$. Since the MS method lacks an encoder model, no reliable estimate for the initial state can be estimated (see Appx. A.2.2). For methods based on RCs, such as [304], a similar warm-up phase is often employed: initial states are usually estimated by iterating the network until time point $\mathbf{x}_t$ by providing ground-truth data $(\mathbf{x}_0 \dots \mathbf{x}_{t-1})$ as input, and then iterating forward with the model-predicted output as input for future time steps.

GEOMETRIC AGREEMENT For evaluating attractor geometries, we use a state space measure D_{stsp} based on the Kullback-Leibler (KL) divergence, which assesses the match between the ground truth distribution of the data $p_{\mathrm{true}}(\boldsymbol{x})$ and the distribution $p_{\mathrm{gen}}(\boldsymbol{x}|\boldsymbol{z})$ freely generated by the inferred DSR model. The probability distributions can be approximated in several different ways from trajectories. Here, we usually sampled long trajectories corresponding to the test set length (usually 100,000 time steps, but sometimes shorter for the empirical time series) from trained systems, removing transients to ensure that the system has reached a limit set (see the discussion of attractors and limit sets in Sect. 2.2). For low-dimensional systems, the KL divergence can be straightforwardly calculated through a discrete binning approximation [54]:

$$D_{\mathrm{stsp}}\left(p_{\mathrm{true}}(\boldsymbol{x}), p_{\mathrm{gen}}(\boldsymbol{x} \mid \boldsymbol{z})\right) \approx \sum_{k=1}^{K} \hat{p}_{\mathrm{true}}^{(k)}(\boldsymbol{x}) \log\left(\frac{\hat{p}_{\mathrm{true}}^{(k)}(\boldsymbol{x})}{\hat{p}_{\mathrm{gen}}^{(k)}(\boldsymbol{x} \mid \boldsymbol{z})}\right), \tag{72}$$

where $K = m^N$ is the total number of bins, with m bins per dimension and N being the dimension of the ground truth system (Fig. 17**b**). A bin number of $m = 30$ per dimension was chosen as a good compromise for distinguishing between successful and bad reconstructions for 3d systems. Since $K = m^N$ scales exponentially with the observation dimension, the number of data required to fill the bins also scales exponentially, and for higher-dimensional systems (e.g. $N = 5$), we reduced the number of bins accordingly [149]. For even higher-dimensional systems ($N > 6$), evaluating D_{stsp} as outlined above is not feasible computationally. We hence resorted to an approximation of the distributions using Gaussian Mixture Models (GMMs). The true data distribution is approximated as a GMM via $p_{true}(\boldsymbol{x}) \approx \frac{1}{T}\sum_{t=1}^{T} \mathcal{N}(\boldsymbol{x}_t, \boldsymbol{\Sigma})$, with Gaussians centered on the observed data points $\boldsymbol{x}_t$ and a covariance $\boldsymbol{\Sigma}$. The generated distribution is similarly calculated as $p_{gen}(\boldsymbol{x}|\boldsymbol{z}) \approx \frac{1}{L}\sum_{l=1}^{L} \mathcal{N}(\hat{\boldsymbol{x}}_l \mid \boldsymbol{z}_l, \boldsymbol{\Sigma})$. The Kullback-Leibler divergence between the two resulting GMMs can then be approximated with an efficient Monte Carlo approximation ([151], see also [212]):

$$\widetilde{D}_{stsp}\big(p_{true}(\boldsymbol{x}), p_{gen}(\boldsymbol{x}|\boldsymbol{z})\big) \approx \frac{1}{n}\sum_{i=1}^{n} \log \frac{1/T\sum_{t=1}^{T} \mathcal{N}(\boldsymbol{x}^{(i)}; \boldsymbol{x}_t, \boldsymbol{\Sigma})}{1/L\sum_{l=1}^{L} \mathcal{N}(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{x}}_l, \boldsymbol{\Sigma})}, \tag{73}$$

where n Monte-Carlo samples $\boldsymbol{x}^{(i)}$ are drawn from the GMM representing p_{true}. The covariance $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$ can be estimated from the encoder model in probabilistic approaches (i.e. SVAE and MTF training). For the deterministic training techniques, such as BPTT-TF, we can also set it to a scaled identity matrix. We found that values around $\sigma^2 = 0.1 - 1.0$ differentiated well between reconstructions, underscored by the observation that the logarithm of the GMM approximation correlated well with approximations computed via the binning method (Fig. 17**c**).

TEMPORAL AGREEMENT To assess temporal agreement, we computed power spectrum correlations (PSC) in [54] (see Table 3), and Hellinger distances D_H in [53, 149, 152] (see Table 4 and Table 5). We first simulated long time series, corresponding in length to the test set length. After standardization, we computed dimension-wise Fast Fourier Transforms (FFT) using `scipy.fft`. The power spectra were smoothened using a Gaussian kernel, and normalized, and the extended, high-frequency tails, which predominantly consisted of noise, were truncated. Illustrations of different values for the smoothing width σ are given in Fig. 17**d**. For the PSC, we computed dimension-wise correlations between the smoothed ground-truth spectra $F(\omega)$, and generated spectrum $G(\omega)$. The Hellinger distance [276], between spectra is given by:

$$H(F(\omega), G(\omega)) = \sqrt{1 - \int_{-\infty}^{\infty} \sqrt{F(\omega)G(\omega)}\,d\omega} \in [0, 1] \tag{74}$$

In both cases, the dimension-wise PSC values and Hellinger distances were then averaged across dimensions to obtain the reported values. The Hellinger distance D_H has the advantage that it is normalized between 0 and 1, and distinguishes better between good and bad reconstruction, while the PSC exhibits a significant ceiling effect, crowding values close to one (see Table 3).

MAXIMUM LYAPUNOV EXPONENT As discussed in Sect. 2.2 and Sect. 3.2.1, the maximum Lyapunov exponent of a system quantifies trajectory divergence

and is computed in the limit of T going to infinity from the product of Jacobians. To approximate the maximum exponent numerically, we first iterated a trained model forward from some initial condition (randomly drawn or by initializing using the encoder on the test set), and discarded transients. Given the product of Jacobians explodes for chaotic systems (Sect. 3.2.1), we employed a numerical algorithm from [410, 424] that re-orthogonalizing the series of Jacobian products at regular intervals using a QR decomposition. For well-known benchmark systems, ground truth values for the maximum Lyapunov exponent are given in the literature (Lorenz: $\lambda_{max} = 0.905$, Rössler: $\lambda_{max} = 0.072$, Alligood, Sauer, and Yorke [9]). For other systems, such as the Lewis-Glass network model ($\lambda_{max} = 0.072$), we approximated the exponent using the Julia library `DynamicalSystems.jl` [82] and the `dysts` Python package [125], both based on the same algorithm from [424]. The values obtained through this method were consistent with values from the literature (Lorenz: $\lambda_{max} = 0.903$, Rössler: $\lambda_{max} = 0.071$).

4.1.2 *Performance Metrics for Multimodal Time Series*

This subsection introduces the multimodal performance metrics used in [53] and for the results in Sect. 4.3.

STATE SPACE MEASURE WITHOUT CONTINUOUS OBSERVATIONS If an underlying continuous DS is only available through discrete observations, a direct mapping between the state space of the reconstructed system and the true DS is generally not available.

Suppose we have temporal alignment between states of the true DS and states of the reconstructed DS, and both systems have the same dimensionality ($M = N$). In that case, we can directly overlay latent states inferred from the encoded data with the ground truth latent states. Assume we have observed an ordinal sampling $p(\boldsymbol{o}_t|\boldsymbol{x}_t)$ of a continuous underlying DS. After training, we can draw states from the encoder model $p(\hat{\boldsymbol{z}}_t|\boldsymbol{o}_t)$ that are temporally aligned with the ground truth states $\boldsymbol{x}_t$ of the continuous DS. Then we can directly optimize a linear matrix $\boldsymbol{B}$, aligning all encoded states $\hat{\boldsymbol{Z}}$ and ground truth states $\boldsymbol{X}$ via linear regression:

$$\boldsymbol{B} = \left(\hat{\boldsymbol{Z}}^\top \hat{\boldsymbol{Z}}\right)^{-1} \hat{\boldsymbol{Z}}^\top \boldsymbol{X} \tag{75}$$

This procedure was used to overlay the reconstructed Lorenz-96 system in Fig. 29.

In the more general case, the dimensionality of the reconstruction model can be higher ($M > N$) than the dimension of the underlying DS, or we do not have access to latent states temporally aligned with the states of the underlying DS. In this case, to still assess to what degree reconstructed systems agree in terms of attractor geometry, we require a mapping between the two state spaces that does not introduce any additional degrees of freedom. To this end, we optimize an operator that consists of a linear projection operation, combined with a rotation operation that preserves geometry, which was either found via grid search over rotation matrices or via Procrustes transformation [273]. The procedure is illustrated in Fig. 18. The linear projection was computed by first applying Principal Component Analysis (PCA) and then re-standardizing all axes. As

a second step, we identified a rotation matrix to rotate the latent state space, aiming to minimize D_{stsp} used for assessing attractor geometries agreement. Fig. 19**a** shows how state space agreement is affected by different rotations. Fig. 19**b** illustrates that for a system trained on continuous observations with a co-trained linear observation model, where values of D_{stsp} can be directly obtained via the binning method described in Sect. 4.1.1, those values given by D_{bin} are strongly correlated with those approximated via D_{PCA} in the reconstructed latent space. While Procrustes analysis is numerically more efficient, we found it less effective despite being less conservative than our method.

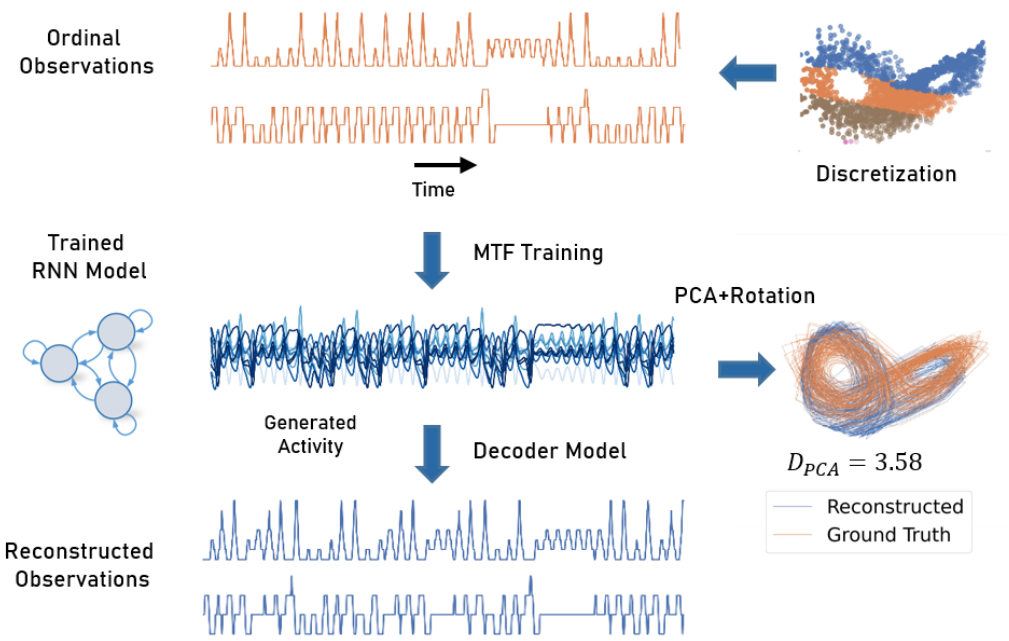

Figure 18: Illustration of the state space measure in the absence of continuous observations.

TEMPORAL AGREEMENT FOR DISCRETE OBSERVATIONS To evaluate the temporal alignment between model-generated and ground truth time series, particularly for *ordinal* and *count* observations, we computed the average Spearman Autocorrelation Function (SACF) up to 200 time lags, using `scipy.stats.spearmanr`, defined by:

$$\mathrm{SACF}_i(\tau) = \frac{\sum_{t=1}^{T-\tau}(r_{i,t} - \bar{r}_i)\cdot(r_{i,t+\tau} - \bar{r}_i)}{\sum_{t=1}^{T}(r_{i,t} - \bar{r}_i)^2}, \tag{76}$$

where $r_{i,t}$ and $r_{i,t+\tau}$ are the ranks of the discrete observations, and $\bar{r}_i$ is the average rank of the time series. Example SACF functions are illustrated in Fig. 24. To obtain a performance error based on this function, we calculated the average squared error between the corresponding SACFs of ground truth and generated time series across all lags and dimensions:

$$\mathrm{MSE}_{SACF} = \frac{1}{D \times T}\sum_{i=1}^{N}\sum_{\tau=1}^{T}\left(\mathrm{SACF}_{\mathrm{gen},i}(\tau) - \mathrm{SACF}_{\mathrm{ground\ truth},d}(\tau)\right)^2 \tag{77}$$

For the values reported in Table 5, OACF corresponds to this error for the ordinal observations, while CACF responds to the error for count observations.

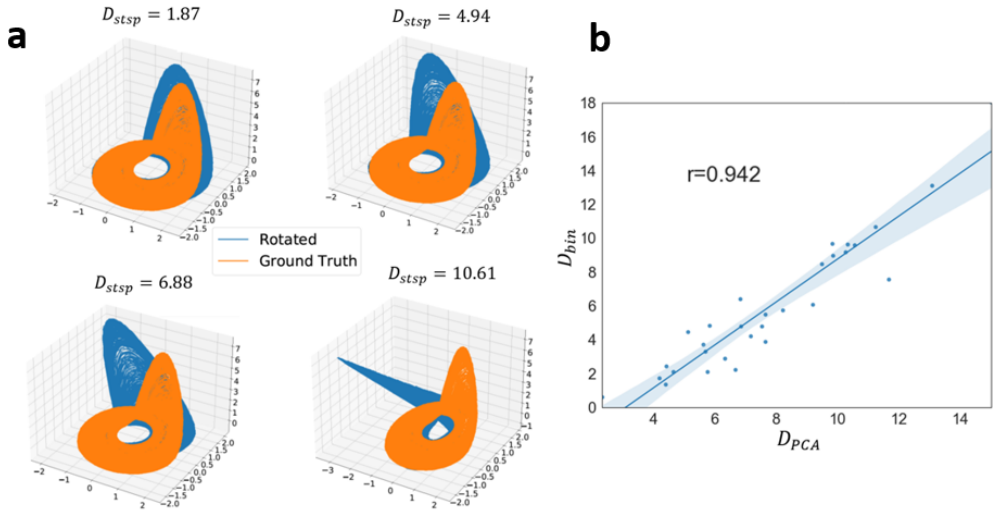

Figure 19: **a**: Ground truth and rotated attractors of the Rössler system with associated D_{stsp}-values. **b**: Correlation between D_{stsp} for models trained on trajectories from the Rössler system, computed directly in observation space given a co-trained linear (Gaussian) observation model (D_{bin}), and after approximation applying PCA and the combined rotation operation directly in the 20-dimensional state space (D_{PCA}), based on a total of 30 trained models. Based on [53].

GLOBAL CORRELATION STRUCTURE OF DISCRETE VARIABLES To determine if reconstructions maintain the global cross-correlation structure among ordinal time series, we computed the Spearman correlation for each pair of ordinal time series, for both the generated and ground truth test data:

$$SCC_{ij} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{r_{it}-\bar{r}_i}{\sigma_{r_i}}\right)\left(\frac{r_{jt}-\bar{r}_j}{\sigma_{r_j}}\right) \tag{78}$$

with σ_{r_i} and σ_{r_j} the standard deviation of the ranks of time series i and j. We then calculated the mean squared error (MSE) across all elements of the respective correlation matrices given N ordinal time series:

$$MSE_{SCC} = \frac{1}{N^2-N}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\left(SCC_{gen,ij} - SCC_{\text{ground truth},ij}\right)^2, \tag{79}$$

PREDICTION ERROR FOR DISCRETE VARIABLES For ordinal variables, a PE was computed analogously to the case of continuous variables. As the MSE is implicitly based on the maximum likelihood principle of Gaussian observations, for non-metric spaces given e.g. by ordinal data, this is not necessarily the best choice. As suggested by Öğretir et al. [450], we here instead take the absolute (L_1) deviation between observed and predicted values:

$$OPE(n) = \frac{1}{N(T-n)}\sum_{t=1}^{T-n}\sum_{i=1}^{N}|o_{i,t+n} - \hat{o}_{i,t+n}| \tag{80}$$

SPIKE STATISTICS FOR COUNT VARIABLES For the hippocampal spike data in Sect. 4.3.3, we evaluated a number of spike statistics. The firing rate for each

neuron i was calculated by dividing the total spike count by the observation time:

$$\mu_i = \frac{\sum_{t=1}^{T} c_{it}}{T}, \tag{81}$$

where c_{it} is the count of neuron i at time t, and T is the total number of time bins. The coefficient of variation (CV) for each neuron i is defined as the ratio of the standard deviation to the mean of spike counts:

$$CV_i = \frac{\sigma(c_i)}{\mu(c_i)}, \tag{82}$$

For the autocorrelation function and cross-correlation function, we used the SACF (Eq. 76) and SCF (Eq. 78) defined above.

4.1.3 *Generalization Error in DSR*

In [139], we developed a combination of a statistical error evaluating agreement in limit sets, and a topological error, aiming to assess the topological equivalence of ground truth and reconstructed systems. Both errors were developed to be sensitive to multistability, but can in principle also be employed to quantify the reconstruction quality of monostable systems.

STATISTICAL ERROR The statistical error is defined using the sliced Wasserstein-1 distance (SW1) [47], between the occupation measures $\mu^{\Phi}_{x,T}$ of a ground-truth DS Φ and $\mu^{\Phi_R}_{x,T}$ of the reconstructed DS Φ_R:

$$SW_1(\mu^{\Phi}_{x,T}, \mu^{\Phi_R}_{x,T}) = \mathbb{E}_{\xi\sim\mathcal{U}(S^{n-1})}\left[W_1(g_\xi\sharp\mu^{\Phi}_{x,T}, g_\xi\sharp\mu^{\Phi_R}_{x,T})\right], \tag{83}$$

where S^{n-1} represents the unit hyper-sphere. The statistical error $\mathcal{E}_{stat}$ is then defined as the integration of SW_1 over initial conditions from a subset U of state space:

$$\mathcal{E}^{U}_{stat}(\Phi_R) := \int_{U\subseteq M} SW_1(\mu^{\Phi}_{x,T}, \mu^{\Phi_R}_{x,T})\, dx, \tag{84}$$

TOPOLOGICAL ERROR Topological equivalence between two DS can not easily be assessed numerically, as this entails approximating the homeomorphism between flows. This approximation, e.g. using invertible NNs, has proven challenging in practice, while the reverse direction, i.e. a failure to approximate the homeomorphism e.g. via invertible NNs does not necessarily show that systems are not topologically equivalent. Hence, we assess agreement in topology between ground truth and reconstructed system based on weaker conditions related to their Lyapunov spectra. The first condition requires the signs of Lyapunov exponents to agree so that $sgn(\lambda_i) = sgn(\lambda^R_i)\ \forall i$. Second, we require that the maximum Lyapunov exponents are close to each other: $|\lambda_n - \lambda^R_n| \,/\, |\lambda_n| < \varepsilon_{\lambda_n}$, where ε_{λ_n} is a tolerance threshhold that can be chosen as a hyperparameter. Lastly, the limit sets should agree in state space. A natural metric to assess this is the Hausdorff distance:

$$d_H(\omega(x, \Phi_R),\, \omega(x, \Phi)) < \varepsilon_{d_H} \tag{85}$$

The Hausdorff distance is motivated by its robustness and sensitivity to outliers between the two sets, but other distance measures are also possible. The topological generalization error then combines all three conditions conditions, using an indicator function $\mathbb{1}\Phi_R(\boldsymbol{x})$ that is only one if all conditions are met:

$$\mathcal{E}_{\text{top}}^{U}(\Phi_R) = 1 - \frac{1}{\text{vol}(U)} \int_{U \subseteq M} \mathbb{1}\Phi_R(\boldsymbol{x}), d\boldsymbol{x}, \tag{86}$$

These errors are sensitive to a failure to reconstruct multistable systems, and the resulting error is proportional to the volume of the basin of the not-reconstructed attractor in a multistable system (see [139] for a formal proof):

$$\mathcal{E}_{\text{gen}}^{M_{\text{test}}}(\Phi_R) \propto \text{vol}(B(A_k)). \tag{87}$$

Here, $\mathcal{E}_{\text{gen}}$ represents both $\mathcal{E}_{\text{top}}$ and $\mathcal{E}_{\text{stat}}$, with Eq. 87 holding for both errors.

LEARNABILITY OF A DS The concept of learnability has been widely discussed in statistical learning theory [363, 405] and deep learning [403]. Assume we have an inference algorithm from a hypothesis class $\mathcal{H}$. In the simplest case, a hypothesis class is called learnable if the error between the learned function and the ground-truth function decreases as more data is provided, and goes towards zero in the limit of infinite data. To make this concept applicable to multistability and out-of-domain generalization in DSR, assume for simplicity that the state space segregates into 2 basins (domains) (see e.g. Fig. 6 bottom row), where one subset is the training domain M_{train} and the other the test domain M_{test}. Assume we have a DSR algorithm $\mathcal{H}$ that includes hypotheses consistent with both the training and test data [29, 402], i.e. models that perfectly approximate the underlying DS on both train and test domain. We define $\Theta_0 = \{\boldsymbol{\theta} \in \Theta | \mathcal{E}_{\text{gen}}^{M_{\text{train}}}(\Phi_\theta) \approx 0\}$ as the set of parameters that have (near) zero reconstruction error on the training domain, with $\mathcal{H}_0$ the corresponding reconstructed DS. Then, we define learnability in DSR as [139]:

Definition 1 *The OODG problem $(\mathcal{H}, \mathcal{D})$ defined by the hypothesis class $\mathcal{H}$ and data set $\mathcal{D}$ is strictly learnable if*

$$\forall\ \Phi_R \in \mathcal{H}_0 : \quad \mathcal{E}_{\text{gen}}^{M_{\text{test}}}(\Phi_R) = 0 \tag{88}$$

Hence, the OODG problem is strictly learnable, if zero reconstruction error on the training domain leads to zero reconstruction error on the test domain.

For hypothesis classes based on universal approximators, there can be infinitely many models in $\mathcal{H}_0$ with different generalization errors on M_{test} (e.g. models that locally overfit M_{train} vs. generalizing models). For a parameterized hypothesis class $\mathcal{H}_\theta = \{\Phi_\theta | \boldsymbol{\theta} \in \Theta \subset \mathbb{R}^P\}$, such as those based on NNs, the optimization procedure contains several stochastic components (initialization, optimization via stochastic gradient descent, drawing from distributions e.g. in VI/MTF), usually leading to a range of values for M_{test}. Given the hypothesis class, $\mathcal{H}_\theta$, we can instead express the distribution over generalization errors for models in $\mathcal{H}_0$.

Definition 2 *We define this as the learnability-distribution of the OODG problem $(\mathcal{H}_\theta, \mathcal{D})$ as*

$$p(\varepsilon_{\text{gen}}|\mathcal{D}) = \frac{1}{\text{vol}(\Theta_0)} \int_{\Theta_0} \mathbb{1}[\mathcal{E}_{\text{gen}}^{M_{\text{test}}}(\Phi_{\boldsymbol{\theta}}) = \varepsilon_{\text{gen}}] d\boldsymbol{\theta}, \tag{89}$$

Here, $\mathbb{1}[\cdot]$ is 1 if the condition in brackets holds and 0 otherwise. If $p(\varepsilon_{\text{gen}}|\mathcal{D})$ is fully concentrated at $\varepsilon_{\text{gen}} = 0$, the OODG problem becomes strictly learnable, while the mass of the distribution around zero quantifies the learnability of the problem. One of the main results in [139] is that for regression-based approaches such as SINDy [59], the strong inductive bias given by a predefined function library containing the correct functions constituting a (potentially multistable) DS $f \in \mathcal{B}_L$ can guarantee that the OODG problem given by $f \in \mathcal{B}_L$ and an observed trajectory $\Gamma_{x_0} \subset \mathcal{D}$ is strictly learnable under some conditions (for more details, see Sect. A.2.1). On the other hand, approaches based on universal approximators, while necessary for DSR from empirical data, will largely fail to generalize to basins not observed in the training data for reasons detailed in more depth in [139].

4.2 RECONSTRUCTIONS FROM UNIMODAL TIME SERIES

4.2.1 *Unimodal Benchmarks*

As discussed in Sect. 2.3 and illustrated in Table 1, DSR algorithms are still overwhelmingly benchmarked on unimodal, monostable DS. This section will first present results on a range of simulated benchmark systems, including a detailed comparison to several other state-of-the-art (SOTA) DSR algorithms, based on the results from [54] and [152]. Details on all benchmark systems are given in Appx. A.3, while comparison methods are described in more detail in Appx. A.2.

BISTABLE WILSON-COWAN MODEL To illustrate the mathematical tractability of the dendPLRNN (Sect. 3.1.1), we first reconstructed the dynamics of a relatively simple 2-dimensional Wilson-Cowan model of a population of both excitatory and inhibitory neurons (see Appx. A.3.1 for details), containing two stable and one unstable equilibrium points (EPs). Reconstructed vector field and EPs closely agree, as illustrated in Fig. 20. The vector field of the learned model was approximated by computing the 1-step difference vectors $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}) - \boldsymbol{z}_n$ induced by applying the trained dendPLRNN across a grid of sample points and decoding the resulting values using the linear observation model. The analytically determined EPs agree both in terms of their location and in terms of their stability, as assessed by the system's Jacobians around the EPs.

BENCHMARK SYSTEMS We then compared the performance of the dendPLRNN, trained with VI and BPTT-TF (using id-TF introduced in Sect. 3.2.3), on four simulated benchmark systems, featuring two low-dimensional examples (Fig. 21) and two high-dimensional examples (Fig. 22). For all reconstructions displayed in these figures, trajectories were sampled by initializing the network at the first time step of the test set, where the initial state was estimated by some mapping $p(\mathbf{z}_0|\mathbf{X})$ as discussed in Sect. 4.1.1, and hence go beyond short-term forecasts, in that the system's temporal and geometric structure need to be fully captured in its dynamics equation.

We first investigated the famous 3D chaotic Lorenz attractor (Lorenz-63) [252], to date the most common benchmark for DSR algorithms. The reconstructions in Fig. 21 show that the dendPLRNN can faithfully reconstruct both the tem-

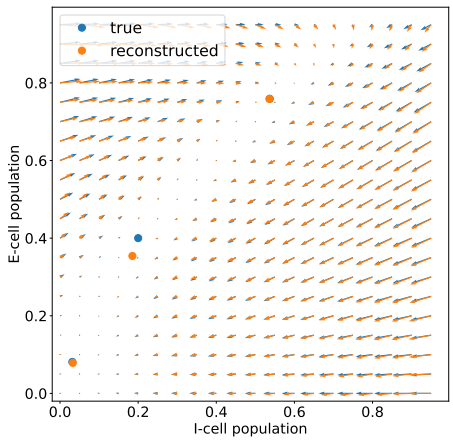

Figure 20: Reconstructed and ground truth vector field for the 2D Wilson-Cowan system, including locations of the analytically obtained fixed points the trained dendPLRNN, and ground truth fixed point locations. Taken from [54].

poral and geometric structure of the true system. The analytically computed fixed points of the reconstructed system, displayed as dots, closely agree with those of the ground truth system. This result is notable in so far as these fixed points are never actually observed during training. This implies that the trained model not only locally fits the observations but captures the global topological structure of the state space.

We then reconstructed a 3d biophysical model of a bursting neuron from [94] (Fig. 21**b**). This model is challenging to reconstruct in that it features two very different time scales: one producing fast spiking behavior, and one producing slow oscillations. Different time scales are challenging to learn due to the often mentioned exploding and vanishing gradient problems when training RNN models (Sect. 3.2.1, [33]), and are often explicitly addressed e.g. by regularization techniques [355] or special network architectures featuring gating mechanisms, such as the LSTM network [155]. While reconstructions with BPTT-TF almost perfectly match the true system, training with VI did not achieve comparably good reconstructions for this model, likely due to the challenges discussed in Sect. 3.2.2.

We then reconstructed two high-dimensional examples: a 10-dimensional spatially extended Lorenz-96 weather model [253] with local interactions (Fig. 22**a**), and a chaotic neural population model [224] with 50 neurons (Fig.22**b**). For both systems, Fig.22 displays ground truth and generated time series plots, spatiotemporal patterns and reconstructed and ground truth power spectra (see Sect. 4.1.1).

BENCHMARK COMPARISONS We then compared the dendPLRNN trained either with VI or with BPTT-TF with four other algorithms, described in more detail in Sect. A.2. First, SINDy [59] reconstructs governing equations by first approximating numerical derivatives. Then, appropriate terms from a library of basis functions, e.g. polynomial or trigonometric functions, are selected by some sparse regression algorithm, e.g. LASSO regression. Second, [408] suggests a hybrid of LSTMs, trained using truncated BPTT, and which approximates a mean-

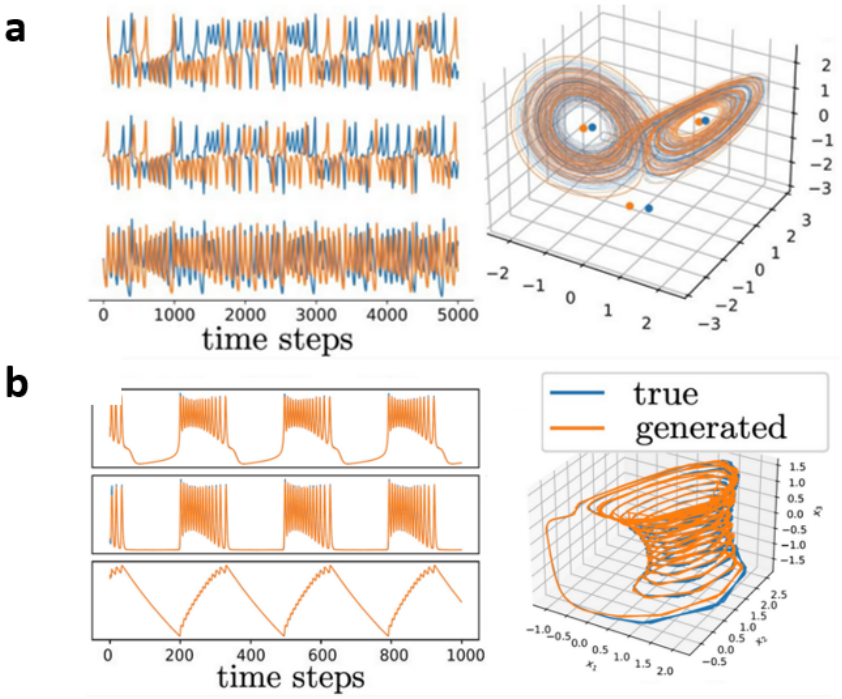

Figure 21: Example reconstructions of two low-dimensional benchmark systems, produced from a dendPLRNN trained with VI ($B = 20$, $M = 15$) on the chaotic Lorenz system (**a**, Eq. 110) and with STF ($B = 47$, $M = 26$, $\tau = 5$) on the bursting neuron model (**b**, Eq. 112), with time series (left) and state space reconstructions for both true and generated time series (right). Since the bursting neuron model is non-chaotic, model predictions agree closely with the ground truth data for up to 1000 time steps, while due to the chaotic nature of the Lorenz system, predictions diverge while agreeing in terms of overall temporal and geometric structure. Taken from [54].

field stochastic model (MSM) based on Ornstein-Uhlenbeck processes (LSTM-MSM) in case a trajectory diverges too far from the training data. Third, as suggested in [303] we use reservoir computing (RC) with reservoir parameters selected to ensure the 'echo state property' [174]. For higher-dimensional systems, as introduced in [303], we use a spatially arranged set of reservoirs with local neighborhood interactions. Finally, we used Neural-ODEs [70], implemented in the `torchdiffeq` package, and using the adjoint method [70] for training. Table 3 summarizes these results. Besides the benchmark systems introduced above, we investigated challenging data situations using the Lorenz-63 system. Here 'low amount of data' denotes short training time series with just 1000 time steps used for training, 'partially observed' implies only training on state variable x in Eq.110, combined with a delay embedding [349, 388] to create a 3d dataset, and the 'high noise' setting combines high process and high observation noise (drawing from a Gaussian with $d\epsilon \sim \mathcal{N}(0, 0.1dt \times \mathbf{I})$ for the process noise and using 10% observation noise, respectively). As indicated by the right column, we chose model sizes to provide roughly the same number of trainable parameters for each model. For important hyperparameters, we performed grid searches to determine optimal settings.

Table 3: Comparison of the dendPLRNN trained by VI or BPTT+TF, RC [303], LSTM-MSM [408], SINDy [59] and Neural ODE ([70]) on 4 DS benchmarks (top) and 3 challenging data situations (bottom). Values are mean ± SEM. Based on [54].

Dataset / Setting	Method	PSC	D_{stsp}	20-step PE	Dynamical variables	Parameters
Lorenz	dendPLRNN+VI	0.997 ± 0.001	0.80 ± 0.25	2.1e−3 ± 0.2e−3	22	1032
	dendPLRNN+TF	0.997 ± 0.002	0.13 ± 0.18	9.2e−5 ± 2.8e−5	22	1032
	RC	0.991 ± 0.001	0.24 ± 0.05	1.2e−2 ± 0.1e−3	345	1053
	LSTM-MSM	0.985 ± 0.004	0.85 ± 0.07	1.2e−2 ± 0.1e−3	29	1035
	SINDy	**0.998 ± 0.0003**	**0.04 ± 0.01**	**6.8e−5 ± 0.2e−5**	3	252
	Neural ODE	0.992 ± 0.001	0.149 ± 0.014	1.1e−3 ± 4.1e−5	3	1011
Bursting Neuron	dendPLRNN+VI	0.55 ± 0.03	7.5 ± 0.4	6.1e−1 ± 0.1e−1	26	2052
	dendPLRNN+TF	**0.76 ± 0.04**	2.9 ± 1.3	6.1e−2 ± 2.2e−2	26	2040
	RC	0.51 ± 0.01	5.1 ± 0.6	8.6e−2 ± 0.1e−2	711	2133
	LSTM-MSM	0.54 ± 0.02	**2.83 ± 0.36**	**3.9e−2 ± 0.1e−2**	45	2166
	SINDy	diverging	diverging	diverging	3	252
	Neural ODE	0.65 ± 0.017	3.85 ± 0.1	2.1e−1 ± 0.5e−2	3	2073
Lorenz-96	dendPLRNN+VI	0.987 ± 0.001	0.10 ± 0.01	3.1e−1 ± 0.9e−1	42	4384
	dendPLRNN+TF	**0.998 ± 0.0001**	**0.04 ± 0.01**	**4.1e−2 ± 0.8e−2**	50	4480
	RC	0.986 ± 0.008	0.25 ± 0.17	7.1e−1 ± 0.1e−2	440	4400
	LSTM-MSM	0.993 ± 0.002	0.23 ± 0.03	8.2e−1 ± 0.3e−2	62	4384
	SINDy	0.996 ± 0.001	0.06 ± 0.003	6.3e−2 ± 0.1e−3	10	27410
	Neural ODE	0.985 ± 0.001	0.21 ± 0.02	4.4e−2 ± 4.5e−3	10	4130
Neural Population Model	dendPLRNN+VI	0.45 ± 0.05	0.56 ± 0.05	**0.82 ± 0.09**	12	821
	dendPLRNN+TF	**0.51 ± 0.01**	**0.19 ± 0.02**	1.53 ± 0.03	75	9990
	RC	0.30 ± 0.05	0.95 ± 0.19	1.82 ± 0.82	50	2500
	LSTM-MSM	0.45 ± 0.03	0.43 ± 0.02	1.02 ± 0.02	56	848
	SINDy	diverging	diverging	diverging	50	66300
	Neural ODE	0.47 ± 0.03	9.56 ± 0.86	**0.58 ± 0.006**	50	10200
Low amount of data	dendPLRNN+VI	0.967 ± 0.007	4.36 ± 0.10	2.8e−2 ± 0.2e−2	22	1032
	dendPLRNN+TF	0.97 ± 0.04	6.9 ± 5.3	1.5e−2 ± 0.9e−2	22	1032
	RC	0.68 ± 0.05	5.74 ± 0.11	4.1e+5 ± 1.2e+5	345	1053
	LSTM-MSM	0.960 ± 0.006	6.06 ± 0.37	2.1e−1 ± 0.3e−2	29	1035
	SINDy	**0.998 ± 0.0003**	**0.04 ± 0.01**	**6.8e−5 ± 0.2e−5**	3	252
	Neural ODE	0.967 ± 0.008	4.66 ± 0.31	1.6e−3 ± 1.8e−4	3	1011
Partially observed	dendPLRNN+VI	0.940 ± 0.006	12.6 ± 1.0	6.5e−2 ± 1.4e−2	22	1032
	dendPLRNN+TF	**0.993** ± 0.003	**0.54** ± 0.16	**5.3e−3** ± 0.2e−3	22	1032
	RC	0.981 ± 0.001	2.92 ± 0.08	7.6e−3 ± 0.1e−3	345	1053
	LSTM-MSM	0.934 ± 0.005	6.06 ± 0.37	2.3e−2 ± 0.3e−2	29	1035
	SINDy	0.974 ± 6×10^{-4}	17.5 ± 0.4	5.1e−2 ± 0.4e−2	3	252
	Neural ODE	0.945 ± 0.004	3.34 ± 0.12	8.3e−3 ± 9e−5	3	1011
High noise	dendPLRNN+VI	0.973 ± 0.006	4.9 ± 0.75	3.5e−2 ± 0.1e−2	22	1032
	dendPLRNN+TF	**0.995 ± 0.002**	**0.4 ± 0.13**	4.6e−3 ± 0.4e−3	22	1032
	RC	0.988 ± 0.001	2.33 ± 0.21	3.1e−2 ± 0.2e−2	345	1053
	LSTM-MSM	0.967 ± 0.006	1.19 ± 0.27	3.3e−2 ± 0.2e−2	29	1035
	SINDy	0.984 ± 0.005	1.01 ± 0.05	**2.3e−3 ± 0.1e−4**	3	252
	Neural ODE	0.982 ± 0.055	0.79 ± 0.06	5.5e−3 ± 1.7e−4	3	1011

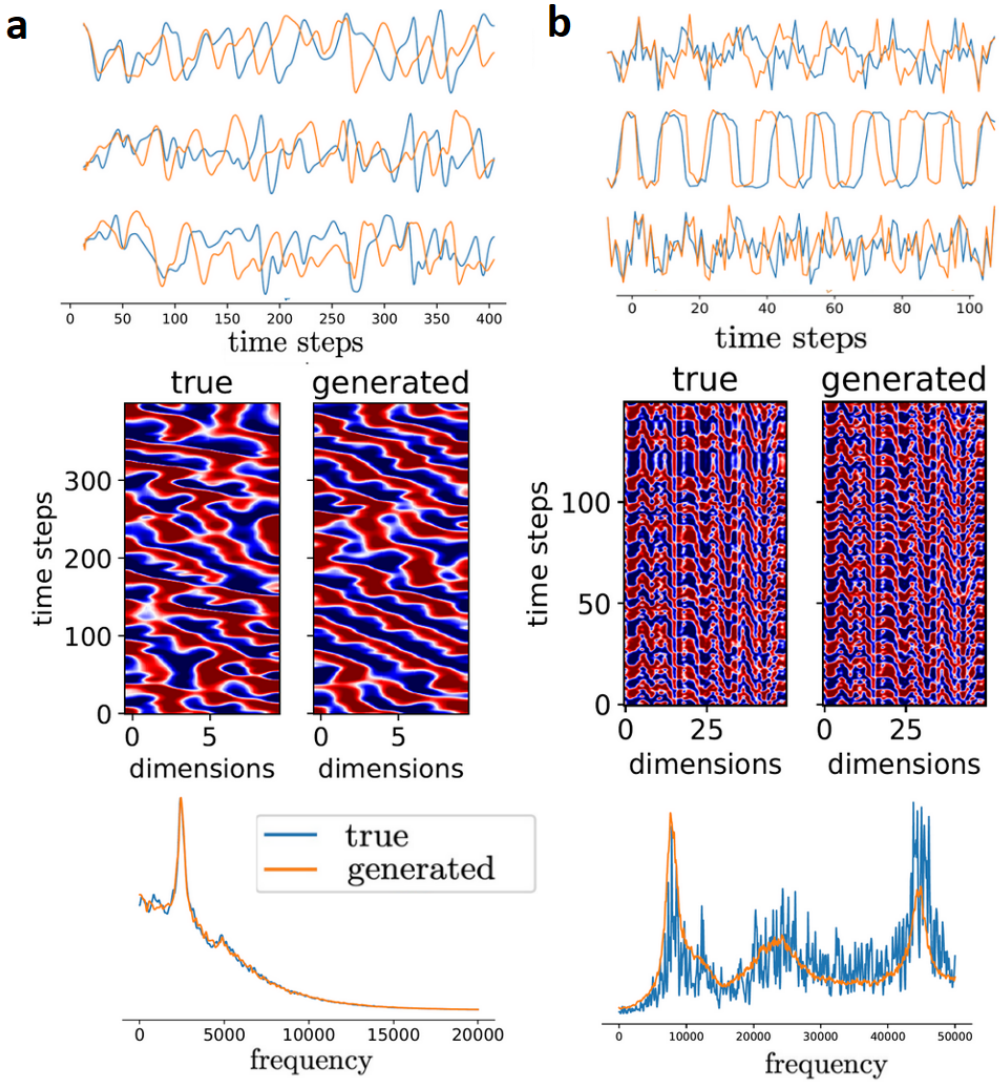

Figure 22: Example reconstruction of DSR from high-dimensional benchmark systems, using a dendPLRNN trained with STF ($B = 50$, $M = 30$, $\tau = 10$). **a**: Time series (top), spatiotemporal evolution (middle), and power spectra (bottom) for the true 10d Lorenz-96 system (Eq. 117) and for time series sampled from the dendPLRNN. **b**: Same for a 50d neural population model (Eq. 119) ($B = 5$, $M = 12$, $\lambda = 1.0$, $M_{reg}/M = 0.2$). Taken from [54].

4.2.2 *Unimodal Experimental Results*

As discussed in 2.3, DSR algorithms are still primarily benchmarked on simulated systems on simulated systems designed as simplified models of complex real-world systems, such as the weather [252, 253]. As laid out in Sect. 2.3, real-world experimental DS are often much more challenging to reconstruct due to a number of factors. Real-world systems, such as the brain, may include a large number of unobserved dynamical variables. Individual neurons ([94], Eq. 112) and even dendrites (Sect. 3.1.1) constitute complex DS in their own right. Moreover, real-world data often contain observation noise and process noise, such as movement artifacts in neural recordings. Real DS can be influenced by unobserved external stimuli, leading to non-stationary behavior, or exhibit very slow time scales. For instance, brain activity is modulated by neurotransmitter or hormone levels varying over weeks and months. Since many DSR algorithms

perform well on simpler benchmarks such as the Lorenz-63 system (see Table 3), comparing them on real-world systems is more revealing due to the aforementioned challenges. Therefore, in [54, 152], we included two real-world DS in our analysis: an Electrocardiogram (ECG) time series, which is a recording of the heart's electrical activity, and an Electroencephalogram (EEG), which measures the electrical activity of the brain by attaching electrodes to the scalp.

Table 4: Comparisons of SOTA DSR algorithms on two challenging experimental datasets (Sect. A.3), adapted from [152]. Values are median ± median absolute deviation over 20 runs. 'dim' refers to the model's dynamical variables, while $|\boldsymbol{\theta}|$ are number of *trainable* parameters. Based on [152].

Dataset	Method	D_{stsp} ↓	D_H ↓	PE(20) ↓	dim	$\|\boldsymbol{\theta}\|$
ECG (5d)	shPLRNN + GTF	$\mathbf{4.3 \pm 0.6}$	$\mathbf{0.34 \pm 0.02}$	$\mathbf{(2.4 \pm 0.1) \cdot 10^{-3}}$	5	2785
	shPLRNN + aGTF	$\mathbf{4.5 \pm 0.4}$	$\mathbf{0.34 \pm 0.02}$	$\mathbf{(2.4 \pm 0.2) \cdot 10^{-3}}$	5	2785
	shPLRNN + STF	7.1 ± 1.8	0.38 ± 0.03	$(5 \pm 2) \cdot 10^{-3}$	5	2785
	dendPLRNN + id-TF	5.8 ± 0.6	0.37 ± 0.06	$(4.0 \pm 0.4) \cdot 10^{-3}$	35	3245
	RC	5.3 ± 1.7	0.39 ± 0.05	$(4 \pm 1) \cdot 10^{-3}$	1000	5000
	LSTM-TBPTT	15.2 ± 0.5	0.73 ± 0.02	$(2.5 \pm 0.5) \cdot 10^{-2}$	70	5920
	SINDy	diverging	diverging	diverging	5	3960
	Neural ODE	12.2 ± 0.7	0.7 ± 0.03	$(4.1 \pm 0.1) \cdot 10^{-1}$	5	4955
	LEM	16.3 ± 0.2	0.56 ± 0.04	$(7.4 \pm 0.1) \cdot 10^{-1}$	62	4872
	Latent ODE	15.1 ± 3	0.61 ± 0.03	$(6.6 \pm 0.2) \cdot 10^{-1}$	5	4852
	ODE-RNN	12.9 ± 1.1	0.67 ± 0.03	$(5.1 \pm 0.1) \cdot 10^{-1}$	5	4816
EEG (64d)	shPLRNN + GTF	$\mathbf{2.1 \pm 0.2}$	$\mathbf{0.11 \pm 0.01}$	$(5.5 \pm 0.1) \cdot 10^{-1}$	16	17952
	shPLRNN + aGTF	$\mathbf{2.4 \pm 0.2}$	$\mathbf{0.13 \pm 0.01}$	$(5.4 \pm 0.6) \cdot 10^{-1}$	16	17952
	shPLRNN + STF	14 ± 7	0.50 ± 0.16	$\mathbf{(2.5 \pm 0.3) \cdot 10^{-1}}$	16	17952
	dendPLRNN + id-TF	3 ± 1	0.13 ± 0.04	$(3.4 \pm 0.1) \cdot 10^{-1}$	105	18099
	RC	14 ± 7	0.54 ± 0.15	$(5.9 \pm 0.3) \cdot 10^{-1}$	448	28672
	LSTM-TBPTT	30 ± 21	0.2 ± 0.1	$(9.2 \pm 2.3) \cdot 10^{-1}$	160	51584
	SINDy	diverging	diverging	diverging	64	133120
	Neural ODE	20 ± 0.5	0.47 ± 0.01	$(5.5 \pm 0.2) \cdot 10^{-1}$	64	17995
	LEM	10.2 ± 1.5	0.38 ± 0.06	$(8.2 \pm 0.6) \cdot 10^{-1}$	76	18304
	Latent ODE	16.1 ± 3	0.47 ± 0.02	$(5.6 \pm 0.2) \cdot 10^{-1}$	64	17915
	ODE-RNN	13.9 ± 2.1	0.59 ± 0.03	$(9.1 \pm 0.6) \cdot 10^{-1}$	64	17859

On both datasets, we compared in total ten different DSR algorithms: the dendPLRNN, trained with id-TF, a shPLRNN trained with GTF, and as before, the LSTM-MSM [408], RCs [303], SINDy [59], three formulations based on Neural ODEs (Neural ODE [70], Latent ODE, and ODE-RNN [336]) and Long-Expressive-Memory (LEM) [341]. For the shPLRNN trained with GTF, we compared results where the optimal parameter for the strength of GTF α was determined by grid search, as well as using the adaptive α scheme described in Sect. 3.2.4. Table 4 shows that both DSR algorithms based on TF outperform all other approaches according to the geometric, temporal and short-term prediction errors introduced in Sect. 4.1. Fig. 23 illustrates this point more directly. Freely generated trajectories of our models iterated forward by only providing some initial state estimate $\mathbf{z}_0$, match the overall behavior of the true time series, while the comparison methods struggle to produce any meaningful long-term patterns or lead to divergences. It is important to emphasize that this point is not merely explained by us mishandling the other approaches or lack of hyperparameter tuning, since comparisons on simulated benchmark systems, such as the Lorenz-63 and Lorenz-96 system, showed performance of the comparison

methods much closer to our methods (see Appx. Table 7). Overall most methods managed to capture long-term statistics of these benchmark systems reasonably well (Appx. Fig. 54). Further, short-term predictions for the comparisons also all looked comparable even on the challenging real-world data (Appx. Fig. 55). In summary, our results show that the reconstruction algorithms based on STF and GTF techniques (Sect. 3.2), combined with the network architectures from Sect. 3.1, are more capable of addressing the challenges inherent in experimental datasets.

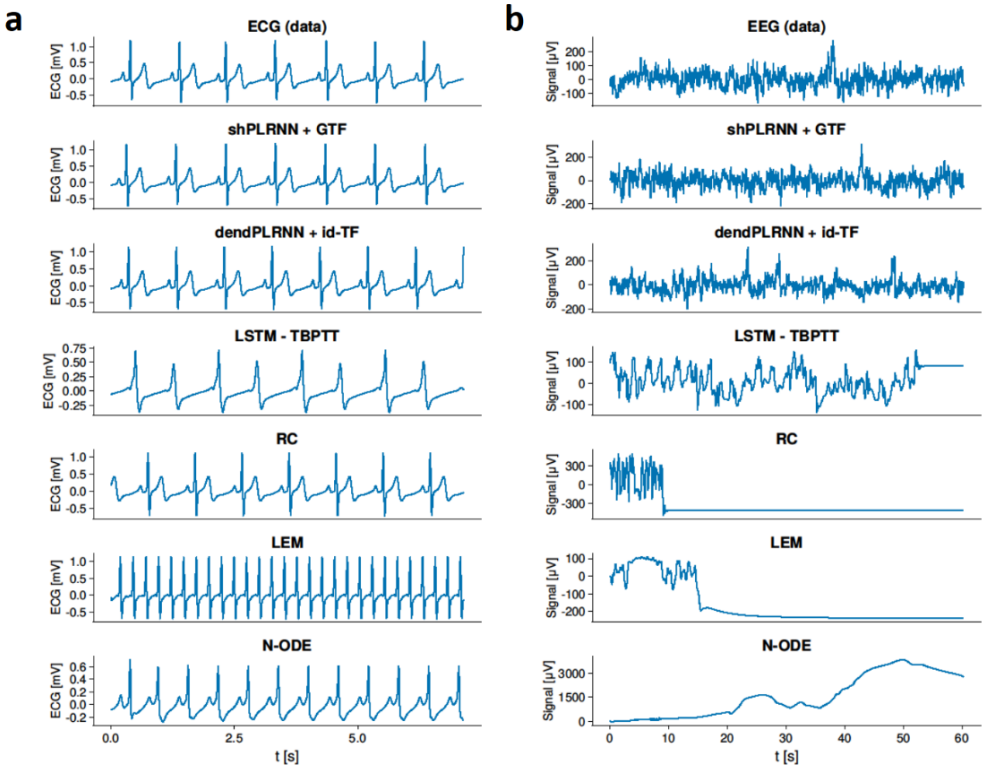

Figure 23: Example time traces of ECG (**a**) and EEG (**b**) reconstructions for all methods compared in Table 4. For each model, we picked the best run out of 20 runs, according to the state space agreement D_{stsp}. Taken from [152].

4.3 RECONSTRUCTIONS FROM DISCRETE AND MULTIMODAL TIME SERIES

This section summarizes the main results from [53], using the MTF framework introduced in Sect. 3.2.5.

4.3.1 *Multimodal Benchmarks*

BENCHMARK COMPARISONS As for the unimodal data, we first began by assessing performance on a range of synthetic benchmark systems. Here we generated training and test datasets from the Lorenz-63 system, the Rössler system, and a 6d Lewis-Glass network model [125, 229] with chaotic dynamics.

To obtain multimodal time series, we sampled ordinal and count observations from these trajectories using the continuous time series as latent states for the ordinal observation model (Eq. 62) and Poisson observation model (Eq. 65) with randomly initialized parameters, and added 10% Gaussian noise to the

continuous observations. Example reconstructions for a dendPLRNN trained by MTF are shown in Fig. 24.

Table 5: Comparison of dendPLRNN trained by MTF, by a multimodal SVAE based on [214], a VAE-TF approach similar to MTF except that all data modalities were 'Gaussianized' (GVAE-TF), BPTT-TF as in [54] using Gaussianized data, and a multiple-shooting (MS) approach. Training was performed on multivariate normal, ordinal, and count data produced by the chaotic Lorenz system, Rössler system, and Lewis-Glass model. Values are mean ± SEM, averaged over 15 trained models. X = value cannot be computed for this model (e.g., because resp. decoder model is not present). SCC (Spearman cross-correlation), OACF (ordinal autocorrelation function), and CACF (count autocorrelation function) all refer to mean-squared-errors (MSEs) between ground truth and generated correlation functions. Bold numbers indicate top performance within ±1 SEM. Taken from [53].

Dataset	Method	D_{stsp} ↓	D_H ↓	PE ↓	OPE ↓	SCC ↓	OACF ↓	CACF ↓
Lorenz	MTF	**3.4 ± 0.35**	**0.30 ± 0.06**	**1.3e−2 ± 2e−4**	**0.12 ± 0.03**	**0.07 ± 0.01**	**0.07 ± 0.01**	**6.6e−5 ± 8.1e−6**
	SVAE	11.1 ± 0.6	0.82 ± 0.05	6.3e−1 ± 5.1e−2	0.68 ± 0.03	0.14 ± 0.01	0.18 ± 0.02	8.5e−5 ± 1.6e−5
	BPTT	6.31 ± 1.2	0.47 ± 0.11	2.1e−1 ± 2.4e−2	0.33 ± 0.04	0.11 ± 0.02	0.09 ± 0.02	8.2e−5 ± 9e−6
	MS	4.5 ± 1.5	0.61 ± 0.08	X	X	0.14 ± 0.04	0.11 ± 0.02	**6.5e−5 ± 3.8e−6**
	GVAE-TF	4.3 ± 0.3	0.47 ± 0.07	3.6e−1 ± 1.5e−3	X	X	X	X
	BPTT-TF	8.8 ± 1.9	0.86 ± 0.05	4.4e−1 ± 2.2e−2	X	X	X	X
Rössler	MTF	**1.45 ± 0.71**	**0.32 ± 0.03**	**1.9e−3 ± 7.1e−5**	**0.08 ± 0.02**	**0.04 ± 0.004**	**0.017 ± 0.003**	**6.5e−5 ± 1.2e−5**
	SVAE	10.7 ± 1.5	0.66 ± 0.05	1.5e−1 ± 3.1e−2	0.24 ± 0.02	0.17 ± 0.03	0.13 ± 0.02	1.1e−4 ± 1.4e−5
	BPTT	9.05 ± 0.5	0.7 ± 0.01	7.4e−2 ± 2.0e−3	0.18 ± 0.02	0.3 ± 0.03	0.19 ± 0.07	1.5e−4 ± 6e−6
	MS	3.99 ± 1.1	0.59 ± 0.04	X	X	0.08 ± 0.04	0.09 ± 0.02	1.6e−4 ± 5.9e−5
	GVAE-TF	12.1 ± 0.5	0.55 ± 0.04	4.9e−2 ± 3.4e−3	X	X	X	X
	BPTT-TF	8.9 ± 1.4	0.64 ± 0.07	2.8e−1 ± 1.8e−3	X	X	X	X
Lewis-Glass	MTF	**0.27 ± 0.07**	**0.33 ± 0.02**	**2.1e−3 ± 7e−5**	**0.11 ± 0.01**	0.12 ± 0.03	**0.05 ± 0.02**	2.3e−4 ± 2.0e−5
	SVAE	2.6 ± 0.5	0.52 ± 0.03	8.0e−2 ± 4e−3	0.26 ± 0.01	0.4 ± 0.05	0.18 ± 0.03	7.5e−3 ± 4.7e−3
	BPTT	2.8 ± 0.5	0.57 ± 0.05	6.2e−2 ± 3e−3	0.23 ± 0.02	0.47 ± 0.08	0.21 ± 0.03	9.1e−3 ± 3.2e−3
	MS	**0.33 ± 0.06**	**0.35 ± 0.01**	X	X	**0.08 ± 0.01**	**0.04 ± 0.002**	**1.9e−4 ± 7.5e−6**
	GVAE-TF	**0.28 ± 0.08**	0.44 ± 0.02	4.6e−3 ± 4e−4	X	X	X	X
	BPTT-TF	2.51 ± 0.71	0.43 ± 0.03	2.6e−2 ± 3e−3	X	X	X	X

We compared performance on these simulated datasets to several other setups for DSR from multimodal time series. While for the evaluations on unimodal continuous time series, several DSR methods existed, this was not the case in the multimodal setting, since, as laid out in 2.4, the problem of multimodal data integration in DSR has hardly been studied, except for the sequential multimodal VAE (MVAE) from Kramer et al. [214]. Besides using the MVAE as a comparison method, we further tried 'classical' RNN training (where observations are provided as external input at every time step, and likelihoods of the observations are computed using modality-specific decoder models), an approach based on 'multiple shooting (MS)' [411], and two methods involving transformation of multimodal data to approximate Gaussian distributions, followed by training the RNN via standard BPTT-TF [54] or VAE-TF without modality-specific decoder models (called 'Gaussianized' VAE-TF, or GVAE-TF). While in the previous comparisons, the DSR algorithm and reconstruction models often went hand in hand (in that RCs, SINDy, or Neural ODEs are all optimized differently and result in different parameterizations of the DSR models), in the comparisons carried out here, we are primarily interested in comparing different training techniques with the same DSR model. We hence used the same DSR model (a dendPLRNN) for all comparisons. As shown in Table 5,

MTF outperformed all other tested training algorithms, often to a significant extent.

Particularly, training with MTF substantially outperformed training with Gaussianized data despite using an otherwise identical algorithm (GVAE-TF). Notably, for the Rössler system, smoothed ordinal and count data led the models to often infer limit cycles instead of chaotic dynamics, leading to much higher D_{stsp} values. Since the Rössler system operates relatively close to a cyclic regime (see also Fig. 33) and its dynamics are only weakly chaotic when for instance compared to the Lorenz-63 system (Lorenz: $\lambda_{max} = 0.905$, Rössler: $\lambda_{max} = 0.072$, [9]), a possible explanation is that smoothing ordinal and count data removes important details in the data that help the algorithm distinguish between cycle and chaotic dynamics, while under-smoothing the data leads to discontinuities in the observations that are difficult to capture.

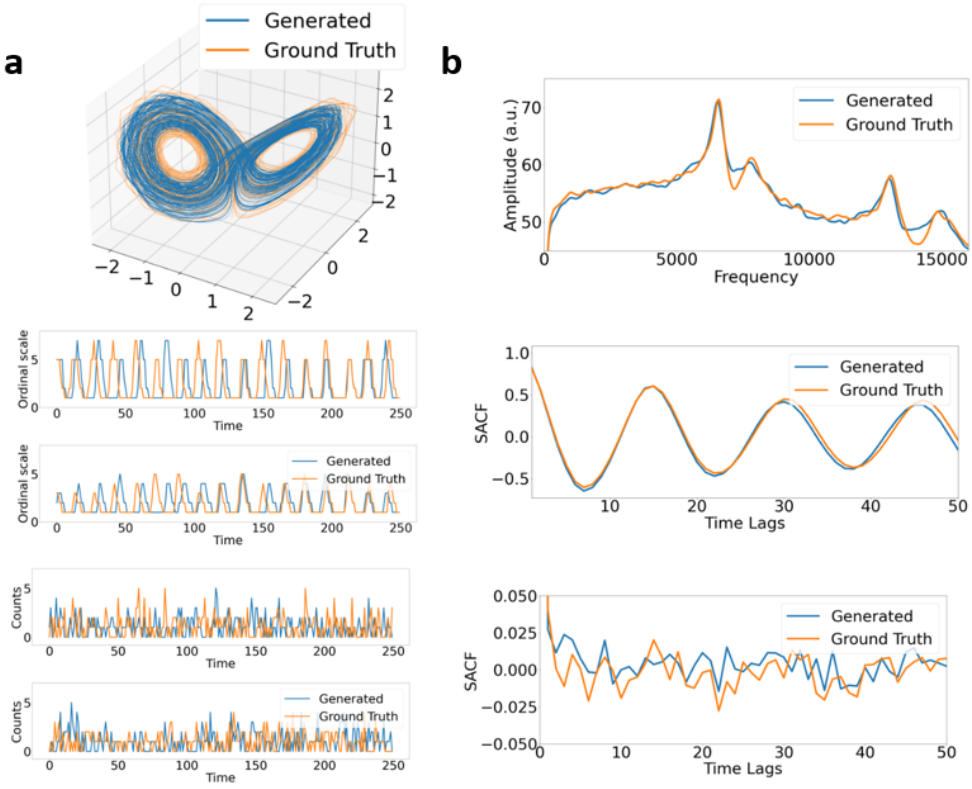

Figure 24: **a**: Sample trajectories and time series produced by a dendPLRNN with parameters ($M = 20$, $B = 10$, $K = 15$, $\tau = 10$), trained using MTF on multimodal data (Gaussian, ordinal, and count)—sampled from a Lorenz-63 system. **b**: Example power spectra from Gaussian data alongside Spearman autocorrelation functions for ordinal and count data. Taken from [53].

RECONSTRUCTIONS FROM HIGHLY NOISY DATA Since in the case of the previous benchmark comparisons, given the moderate level of observation noise added to the continuous observations, DSR is in principle feasible without relying on multimodal data integration, we next explored DSR in which continuous observations are significantly distorted by noise. Here we added observation noise with 50% of the data variance and included ordinal observations with 8 variables divided into 7 ordinal levels, $o_{nt} \in \{1 \dots 7\}$, $n = 1 \dots 8$, again using a randomly initialized ordinal observation model. This subdivides the state space

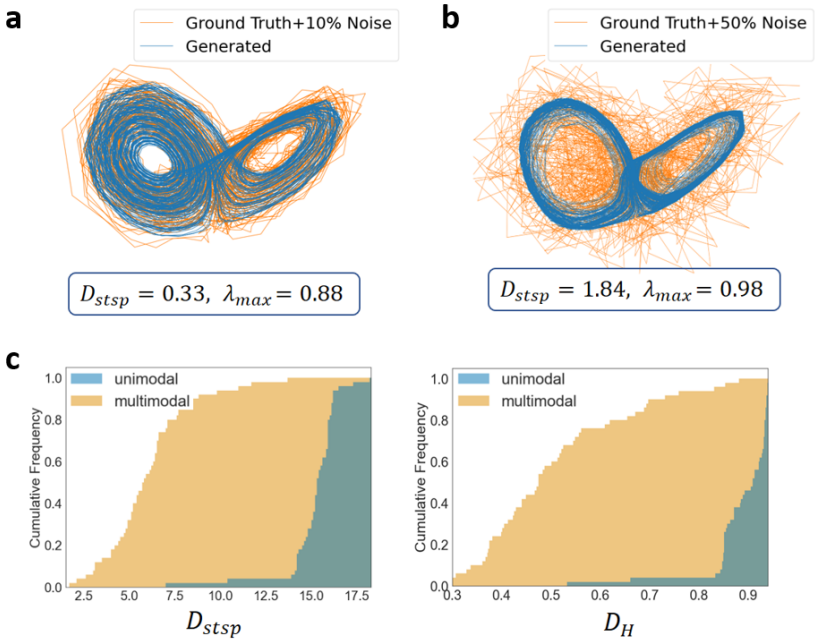

Figure 25: DSR from multimodal observations (continuous and ordinal, sampled from a Lorenz-63 system) using MTF, where continuous observations were distorted with medium (10% of data variance) (**a**) and high (50% of data variance) noise levels (**b**-**c**). **a**: Freely generated example trajectories from a dendPLRNN ($M = 20, B = 10, K = 20, \tau = 10$). **b**: Same as (**a**), but for heavily distorted Gaussian observations. The maximum Lyapunov exponent (λ_{max}) of the dendPLRNN resembles that of the GT system, $\lambda_{max} \approx 0.903$). **c**: Normalized cumulative histograms of geometrical attractor disagreement (D_{stsp}, left) and Hellinger distance (D_H, right) between reconstructed and ground-truth system with and without ordinal observations indicate that DSR from highly distorted data in the unimodal case is impossible. Taken from [53].

into regions belonging to different ordinal levels, which significantly coarse-graining dynamics, especially when different levels occupy larger regions of state space than others (see for instance level 1 in Fig. 28). Example reconstructions are shown in Fig. 25. While successful DSR is next to impossible without including ordinal information, as assessed by the state space measure D_{stsp} and temporal measure D_H (Fig. 25**c**), including ordinal observation allows DSR even under these challenging conditions, and further often successfully captures the chaotic nature of the underlying DS ($\lambda_{max}^{GT} = 0.903$, $\lambda_{max}^{rec} = 0.98$).

CROSS-MODAL INFERENCE FOR MISSING OBSERVATIONS Dealing with missing observations is in principle straightforward in the MTF framework. Missing time steps can be dropped from the respective likelihood terms of the decoder models (Eqs. 53 and 54). For the encoder model, while performing on average worse than the full temporal CNN encoder in situations when all modalities are observed at all time steps (Table 2), the mixture of experts posterior [427] described in Sect. 3.2.6 becomes useful when observations are missing from only one of the channels simultaneously since it combines estimates from both modalities independently into the posterior. We illustrated this by reconstructing the Lorenz-63 system jointly from Gaussian and ordinal observations, where 20% of time steps were missing at random time points individually drawn for each modality (Fig 26a). Fig 26b shows that the MTF framework not only allows

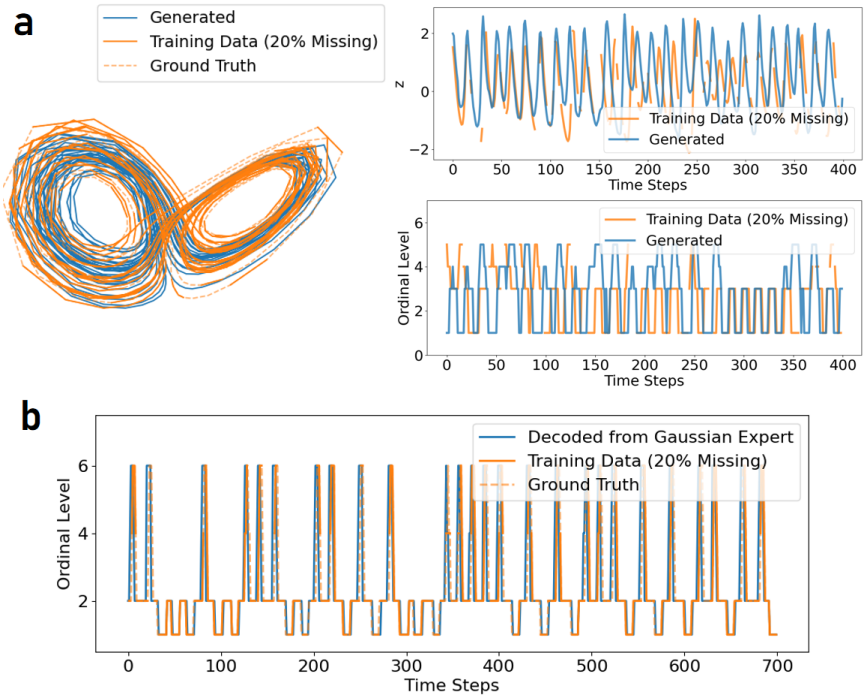

Figure 26: Cross-modal inference, using the mixture-of-experts encoder model. **a**: Reconstructions of the Lorenz-63 from Gaussian and ordinal observations, where 20% of time steps are missing at random time points individually drawn for each modality. This allows the model to develop useful cross-modal links and infer an approximate posterior estimate even when observations from the other modality are missing. **b**: Using only the Gaussian expert to encode ground truth Gaussian observations, the corresponding ordinal ratings can be almost perfectly decoded, including steps missing in the ordinal training data.

successful reconstructions in this challenging situation but learns cross-modal links that allow the inference of the correct ordinal ratings decoded only from the Gaussian expert, including at time points that were missing during training.

RECONSTRUCTIONS FROM PARTIAL OBSERVATIONS Besides integrating across several modalities, the encoder model in the MTF framework more generally facilitates the learning of complex relationships between observations and underlying DS, and can also be used to embed and unfold DS in a higher-dimensional embedding space. As highlighted in the introduction (Sect. 2.3, Fig. 4), temporal delay embeddings (TDEs) are closely related to observation functions, and provide a mechanism to reconstruct and unfold attractors even from partial observations. Given that the CNN encoder employs temporal convolutions, it is in principle well-equipped to learn TDEs directly from the data. This is illustrated in Fig. 27. When training a shPLRNN ($M = 3$) with MTF on one-dimensional partial observations of the Lorenz-63 system (x-coordinate), the encoder model succeeds in unfolding the attractor. The encoded states $p(\tilde{\boldsymbol{Z}}|\boldsymbol{X})$ closely resemble the unfolded attractor using a TDE with settings determined by the first minimum of the mutual information [2, 187]. These capabilities might contribute an explanation as to why the temporal CNN Encoder outperformed all other encoder models tested (Table 2), and underscores why even in the case of contin-

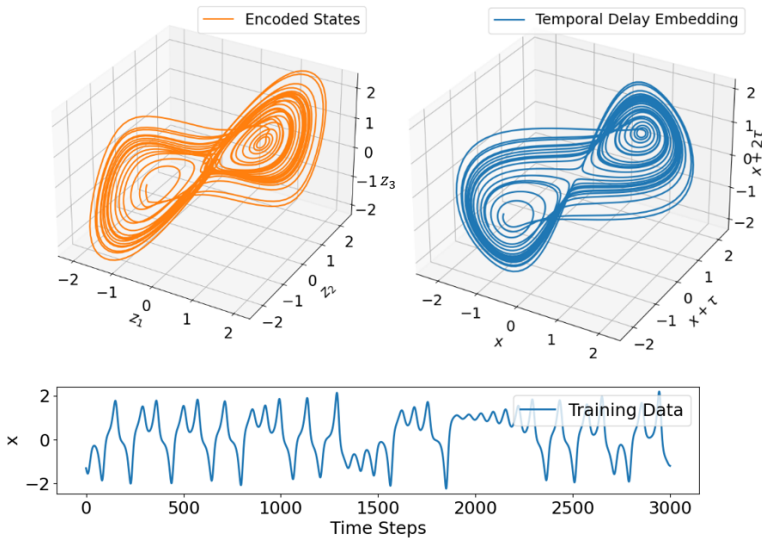

Figure 27: Encoded states $p(\tilde{\boldsymbol{Z}}|\boldsymbol{X})$ (left) after training a shPLRNN with MTF on one-dimensional observations of the Lorenz-63 system (x coordinate, bottom), resemble the unfolded attractor using a temporal delay embedding (right) with optimal settings determined using the minimum of the mutual information [187]. For this plot, states were again approximately overlayed with the delay-embedded states using Eq. 75.

uous and unimodal observations, the MTF framework can provide significant advantages.

4.3.2 *DSR from Discrete Random Variables*

Since the integration of discrete variables on top of distorted continuous observations allows DSR even in situations where it would otherwise be infeasible, we omitted continuous observations altogether and attempted DSR solely from discrete random variables. Assessing to which extent DSR was successful in this context is more challenging, since a priori no mapping to the space of the true underlying DS, as usually provided by the linear Gaussian/identity observation model (Eqs. 14 and 37) exists. To still obtain an estimate for whether the underlying DS is captured, we approximate this mapping after training via a linear dimensionality reduction technique, based on principal component analysis (PCA), combined with a rotation operation that maximizes attractor overlap. This procedure is illustrated in Fig. 18 and allows us to approximate the state space measure D_{stsp} in the reduced space via D_{PCA}. For the ordinal encodings, we subdivided the state space of the underlying DS by randomly initializing an ordinal observation model for each variable, as depicted on the left of Fig. 28, sampling 8 ordinal variables with 7 levels and 60 variables with 2 levels for the reconstructions shown in the bottom row in Fig. 28. While ordinal encodings discretize the underlying dynamical process, particularly when using only two levels per variable as in the bottom row, they maintain the ordering between levels, reflecting the assumption that ordinal levels reflect a continuous underlying variable. Consequently, the ordinal encodings preserve aspects of the original continuous variable's structure.

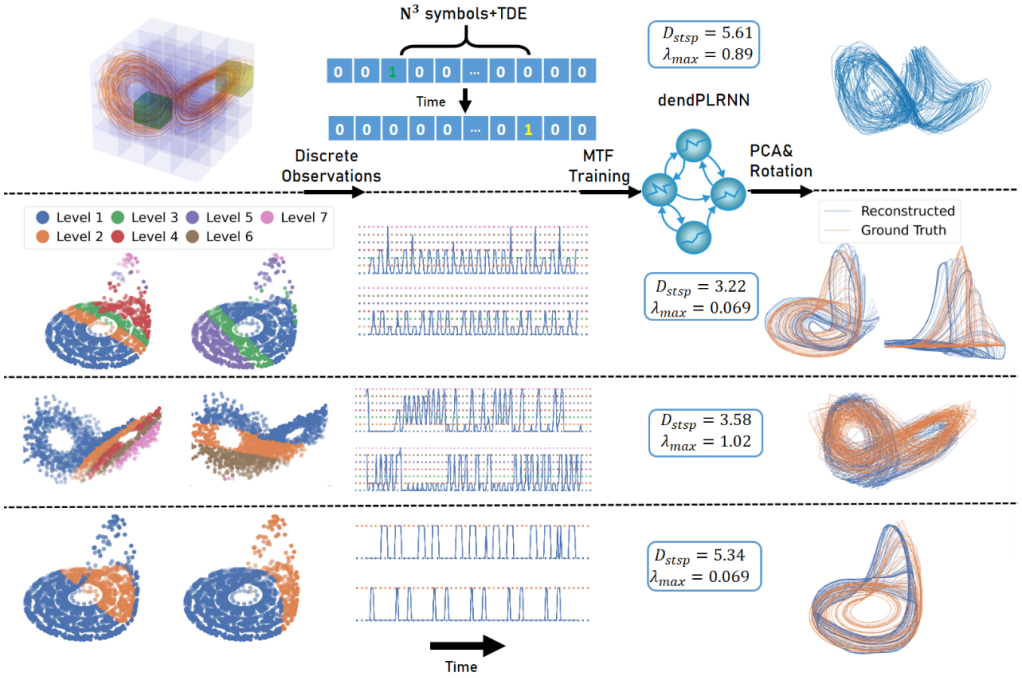

Figure 28: DSR from discrete observations for the Rössler system ($\lambda_{\text{max}}^{\text{true}} \approx 0.072$) and Lorenz system $\lambda_{\text{max}}^{\text{true}} \approx 0.903$). Note that in all cases the topology and general geometry are preserved, and maximal Lyapunov exponents closely match those of the true systems. First row: Symbolic coding of Lorenz attractor (see Fig. 30 for true and predicted class label probabilities and statistics), TDE = temporal delay embedding. Second row: Reconstruction of Rössler attractor from 8 ordinal time series with 7 levels each. Third row: DSR of Lorenz attractor from 8 ordinal time series with 7 levels each. Fourth row: DSR of Rössler attractor from 60 ordinal time series with 2 levels each. Based on [53].

To test our approach in higher dimensions, we reconstructed the 10-dimensional chaotic Lorenz-96 system (Eq. 117) from 30 ordinal variables, each having up to 15 levels. On average, each variable only occupied 7 unique categories due to the random initialization of the observation model. For these reconstructions, we used the shPLRNN (Eq. 18), since the results in [152] (see Table 7) indicate that with the shPLRNN, we can in principle reconstruct the Lorenz-96 system directly in the observation dimension. Given in this situation the reconstruction model has the same state space dimension $M = 10$ as the underlying system, this has the further advantage that when analyzing the reconstructed latent activity we do not require any additional dimensionality reduction techniques. Despite the ordinal sampling leading to a completely different representation of the dynamics of the underlying Lorenz-96 system (Fig. 29**a**), MTF allows us to almost perfectly decode the ground truth Lorenz-96 trajectories from the discrete ordinal time series (Fig. 29**b** middle, **c** right). It is important to emphasize that the model has never seen the original Gaussian observations during training. They are instead only indirectly inferred from the ordinal observations by forcing the DSR model to approximate the underlying DS in its latent space. Further, trajectories freely generated using the DSR model capture the overall complex chaotic spatiotemporal pattern of the Lorenz-96 system well (Fig. 29**b** bottom). Note that here we do not expect perfect alignment of the generated

trajectories with the ground truth data due to exponential trajectory divergence in chaotic systems.

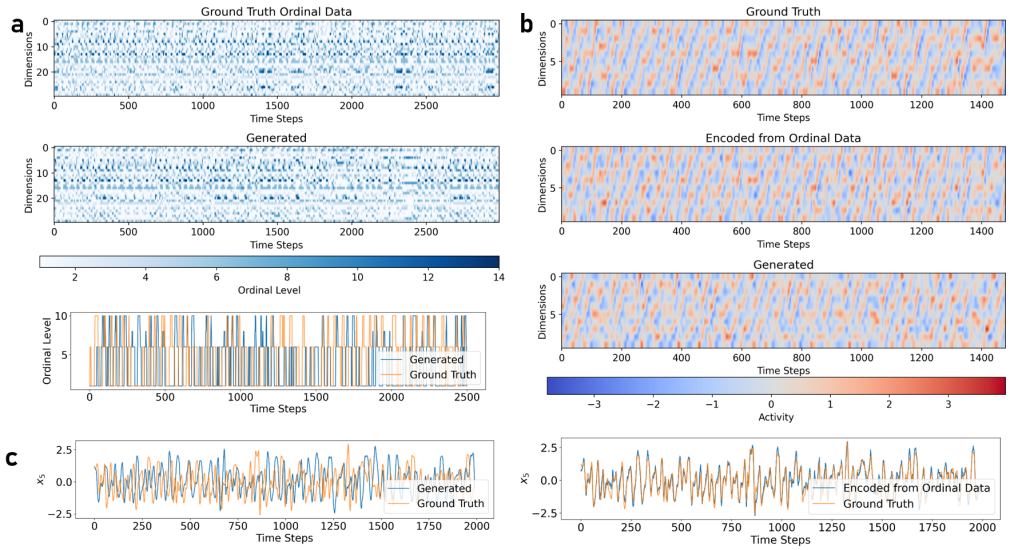

Figure 29: Reconstruction of a 10d chaotic Lorenz-96 system solely from ordinal observations with up to 15 levels using a shPLRNN ([152]; $M = 10, L = 100, \tau = 10$). **a** top: Ground truth ordinal time series sampled from a randomly initialized ordinal observation model $p(\boldsymbol{o}_t|\boldsymbol{x}_t)$ from ground truth states $\boldsymbol{x}_t$ of the Lorenz-96 system, and reconstructed ordinal observations decoded from freely generated latent states using the trained decoder model $p(\boldsymbol{o}_t|\tilde{\boldsymbol{z}}_t)$. Bottom: Example ground truth and freely generated ordinal time series from 1 channel. **b**: Ground truth states $\boldsymbol{x}_t$ (top), states encoded using the trained MTF encoder $p(\tilde{\boldsymbol{z}}_t|\boldsymbol{o}_t)$ (center), and *freely generated* latent activity from the trained DSR model $\boldsymbol{z}_t = F_\theta(\boldsymbol{z}_{t-1})$ (bottom). States $\tilde{\boldsymbol{z}}_t$ encoded from the ordinal data were aligned with ground truth states $\boldsymbol{x}_t$ (not seen during training) using a linear operator $\boldsymbol{B}$ (Eq. 75). This linear operator was also used to project the freely generated activity of the shPLRNN into the observation space of the Lorenz-96 system. **c** left: Example of ground truth (orange) and freely generated (blue) activity. **c** right: Aligned ground truth ($\boldsymbol{x}_t$) and encoded latent states ($\tilde{\boldsymbol{z}}_t$) as in **b** for one example unit. Note that the encoded states $p(\tilde{\boldsymbol{z}}_t|\boldsymbol{o}_t)$ and ground truth states $\boldsymbol{x}_t$ overlap almost perfectly, although the $\boldsymbol{x}_t$ have never been seen by the model during training.

Finally, we explored the feasibility of DSR based on a purely *symbolic representation* of the dynamics. Here, we used 4^3 distinct symbols corresponding to subregions delineated by a $4 \times 4 \times 4$ grid overlaying the attractor. Since 28 of the subregions thus obtained were never visited, we further reduced the symbolic code to 36 independent symbols. Other symbolic encodings are feasible (see Appx. Fig. 56) The symbolic encoding, unlike ordinal encodings, does not inherently retain the structural relationships of the continuous state space. Each symbol in this method represents a specific subregion independently, with all categorical probabilities in Eq. 64 learned separately. Different symbols also occurred with different frequencies, further reducing the information content of the symbolic encoding. However, as shown in Fig. 28 (top) and Fig. 30**a**, reconstructions could be achieved even from this symbolic encoding. Notably, many MTF-trained runs approximated the maximum Lyapunov exponent of the true Lorenz system ($\lambda_{max} = 0.903$) closely, demonstrating that they faithfully captured the system's chaotic nature despite the highly challenging nature of the

representation (Fig. 30**b**). In comparison, other training algorithms like MS or the sequential MVAE [214] either significantly deviated in estimating λ_{max} (MS) or failed to capture meaningful geometric structure (sequential MVAE). Further numerical results underpinning these conclusions for various datasets and settings can be found in Appx. Table 8.

These results underscore the MTF's proficiency in leveraging its flexible encoder-decoder structure to generate TF signals even from data representations substantially different from the Gaussian continuous case. To our knowledge, such reconstructions from symbolic time series of chaotic DS have never been shown before. These applications are interesting from a theoretical perspective, bridging the gap to the field of symbolic dynamics, which studies, among other things, the relationship between symbolic encodings of a DS and its properties, such as topological invariants and spectral properties [162, 268, 294, 442]. However, these advances also have other practical implications. Experimental time series in fields like physics, neuroscience and psychology often exist solely as discrete data (see Sect. 2.7). An application of this method to discrete ordinal data, derived from a psychological study of social interactions, is more comprehensively discussed in Chapter 5.

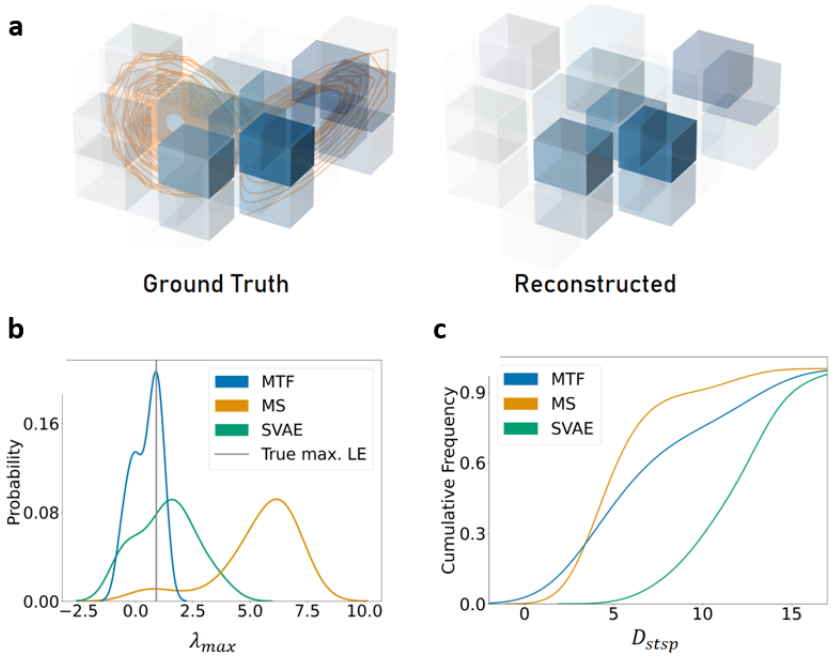

Figure 30: **a**: True and predicted class label probabilities (given the maximum posterior probability for a category at each time step) from a freely generated trajectory of and dendPLRNN, trained with MTF on the symbolic representation of the chaotic Lorenz-63 dynamics. **b** and **c**: Kernel-density estimates of maximum Lyapunov exponents (**b**) and cumulative distributions of D_{stsp} (**c**), comparing training with MTF, Multiple Shooting (MS), and sequential multimodal VAE (MVAE) across 30 trained models each. Taken from [53].

4.3.3 *Multimodal Experimental Results*

We then assessed the MTF's efficacy on two real-world multimodal examples from neuroscience, studying to which extent multimodal integration enhances

reconstructions and builds cross-modal links in the reconstructed system's latent space.

NEUROIMAGING AND BEHAVIORAL DATA As a first experimental baseline, we used recordings from subjects performing cognitive tasks in an fMRI scanner [209]. The dataset included continuous BOLD time series from 26 subjects (sampled at 1/3 Hz) and categorical time series of cognitive stage labels corresponding to five different task stages: ('Rest', 'Instruction', 'Choice Reaction Task [CRT]', 'Continuous Delayed Response Task [CDRT]' and 'Continuous Matching Task [CMT]'). The recorded time series were pre-processed by extracting the first principal component of BOLD activity from 20 brain regions per subject, with each time series spanning $T = 360$ time points. For MTF training, we combined a linear Gaussian observation model (Eq. 50) for the fMRI data, and a categorical observation model (Eq. 64) for the five task stages. We trained models on the first four repetitions of the trial, featuring a total of $T_{train} = 288$ time steps per subject, and held back a test set of length $T_{test} = 72$ to assess short-term predictions. Given invariant geometric and temporal properties can not be meaningfully assessed on very short (test) time series, we computed them along the combined training and test set. This still requires the model to recapitulate the correct long-term properties after only providing one estimated initial state, a challenge most comparison methods struggled with (as also observed on other experimental datasets, Fig. 23).

We then followed a similar analysis to that performed by Kramer et al. [214]. We trained a dendPLRNN using MTF on BOLD signals alone and then by including additional categorical data. Results showed a significant improvement in DSR when incorporating multimodal data, as assessed by comparing geometric and temporal agreement, averaged across 15 runs per subject, and across 20 subjects who did not feature strong movement artifacts in the recorded time series (Fig. 31**a**; paired t-test: D_{stsp}, $t_{19} = 2.45, p < .013$; D_H, $t_{19} = 2.72, p < .007$). This result confirms that categorical data enhances empirical DS reconstruction. The effect was particularly pronounced in the temporal domain, indicating that the categorical variables help structure the reconstructed neural activity in line with the different cognitive task stages. Example short-term predictions on the left-out test set closely matched the real data for some subjects (Fig. 31**b**). Additionally, the latent space after multimodal training showed clear cross-modal links (Fig. 31**c**). Finally, freely generated model predictions after training resemble the overall complex activity pattern of the ground truth data (Fig. 31**d**). Comparative performance metrics are shown in Table 6, with MTF outperforming the sequential MVAE from [214] by significant margins. While other approaches trained via TF methods all performed relatively well on this dataset, including BPTT-TF as described in Sect. 3.2.3 on 'Gaussianized' data, incorporating categorical data via the categorical observation model leads to better performance. Overall, these results underscore two key observations from the previous sections: firstly, that approaches based on TF play a crucial role in achieving successful DSR on challenging experimental datasets, and secondly, that including multimodal information via modality-specific decoder models tailored to their underlying probabilistic assumptions leads to improved reconstructions.

SPIKE TRAINS AND CONTINUOUS POSITION DATA In a second empirical test for the MTF approach, we trained on electrophysiological recordings from

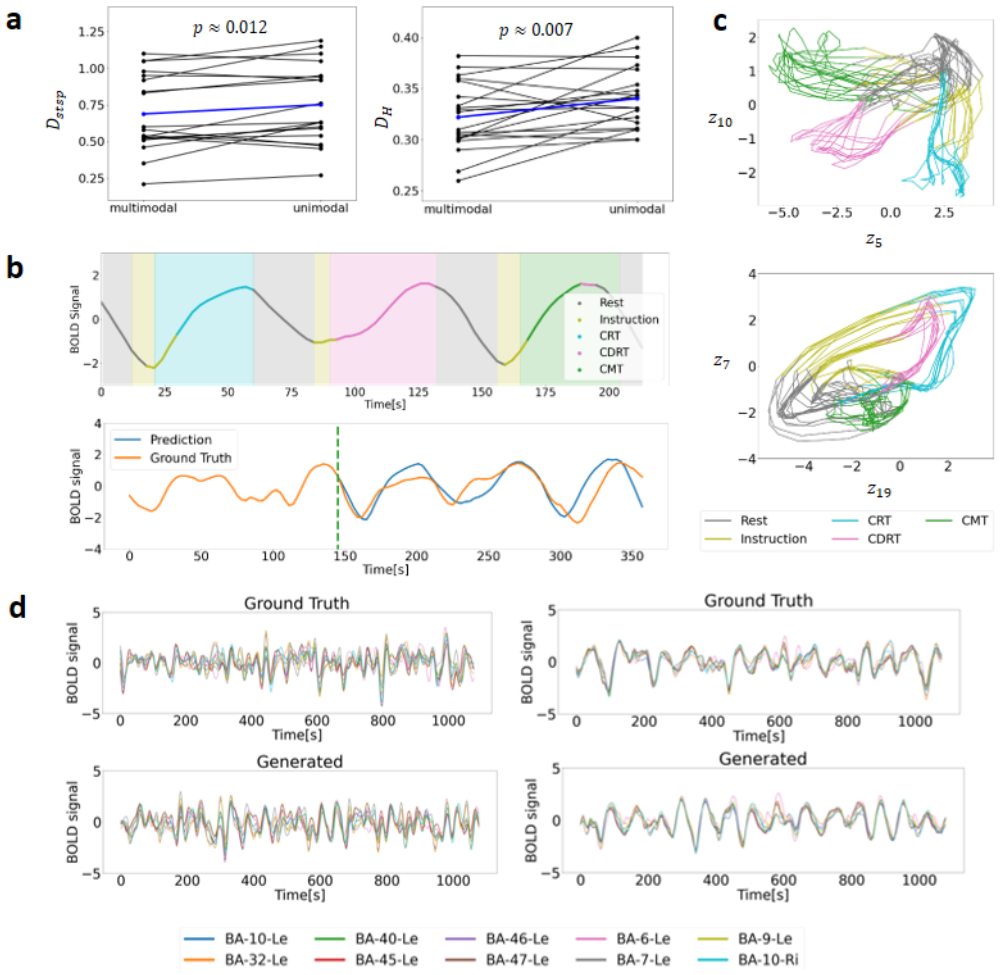

Figure 31: **a**: Multimodal integration on functional magnetic resonance imaging (fMRI)+behavioral data significantly improves DSR compared to just training on fMRI data alone (unimodal). Results are shown for 20 subjects (subjects represented by black lines, with the mean across subjects by a blue line), shown for both geometrical (D_{stsp}, left) and temporal (D_H, right) disagreement between true and reconstructed systems. p-values obtained by performing a paired t-test. **b**: Example of decoded (color-coding of time series) and true (background colors) task stages $\hat{l} \in$ {Rest, Instruction, CRT, CDRT, CMT} for an example subject. The trained model was freely iterated forward from the first time step of the test set not seen during training, and task stages were decoded from the simulated activity based on the maximum posterior probability, $\hat{l}_t = \arg\max p(l_{kt}|z_t)$, given the latent trajectory z_t. **c**: Example subspaces of freely generated latent activity for a DSR model trained jointly on continuous and categorical data by MTF for an example subject. Task labels at each latent state are predicted according to the maximum posterior probability given the latent state at each time step, as in **b**. The latent space is structured according to the task stages. **d**: Freely generated time series from 10 brain areas per subject from subjects #3 (left) and #7. (right). The trained DSR model, only iterated by providing an initial state, captures the overall temporal structure of the complex activity patterns even from very short time series. Based on [53].

the hippocampus of rats navigating a track, provided to the model as spike counts, combined with their longitudinal position data [133]. The hippocampus is a popular area of study of multimodality, given its involvement in spatial

Table 6: Comparison among multi-modal reconstruction methods for experimental fMRI+behavioral data. For each subject and training method, medians across 15 trained models were first obtained for each measure, which were then averaged across 20 subjects (± SEM). SEM = standard error of the mean. X = value not accessible for this method. The abbreviations are the same as in Table 5. Taken from [53].

Dataset	Method	D_{stsp} ↓	D_H ↓	PE ↓
fMRI	MTF	**0.55 ± 0.04**	**0.301 ± 0.007**	**1.21 ± 0.08**
	SVAE	1.9 ± 0.22	0.441 ± 0.019	2.34 ± 0.12
	BPTT	3.31 ± 0.8	0.52 ± 0.05	2.8 ± 0.15
	MS	1.06 ± 0.14	0.373 ± 0.012	X
	GVAE-TF	0.67 ± 0.06	0.335 ± 0.011	1.64 ± 0.07
	BPTT-TF	0.63 ± 0.03	0.312 ± 0.006	1.39 ± 0.05

navigation and memory [62, 284]. The same dataset was recently used to discover joint multimodal embeddings of behavioral and spike data [356], without however extracting any DSR model from the data.

Before training, we first segmented spike trains into 200ms bins [445], obtaining a multivariate count time series. Since many of the 120 recorded neurons only featured very sparse activity, we further filtered the 60 most active neurons from this dataset. We modeled counts using three different decoder models: a standard Poisson decoder, a zero-inflated Poisson decoder, and a negative-binomial decoder (see Sect. 3.2.7 for details). We found the negative-binomial decoder to lead to the best results, whereas the Poisson decoder underestimated and the zero-inflated Poisson decoder overestimated zero counts. Since the negative-binomial model is tailored to deal with high dispersion in count data, it performed well given the high dispersion in the observed time series. The position data was provided as a Gaussian 1d time series. The water reward cues provided at the end of each track were further given to the model as short external reward cues.

We split the overall trial of the rat navigating the maze into equal-sized training and test sets, each comprising 4600 time steps. The model's performance was evaluated by comparing various spike train statistics (mean firing rate, mean zero rate, coefficient of variation, and neural cross-correlation matrix, introduced in Sect. 4.1.2), between the true and generated spike trains across the test set. As shown in Fig. 32**b**, these agreed as closely with the test data as similar comparisons computed between the two sections (training and test set) of the experimental data. Predictions of the rat's position on the test set further matched the real behavior (Fig. 32**a** bottom). Reconstructions on the training set almost perfectly agree with the true data, and are displayed in Appx. Fig. 57.

Crucially, the integration of both spike train and positional data led to more robust and accurate reconstructions than using spike data alone, as confirmed by a Mann-Whitney U-test ($p < 0.025$ for all metrics) across 50 models (Fig. 32**c**, with spike statistics as in Sect. 4.1.2). Fig. 32**d** shows that the model constructs a joint latent embedding of neural activity and the rat's activity movement.

Cells in the CA1 region of the hippocampus are known for their role in encoding spatial information [291]. These cells activate in response to specific locations in an environment, providing a neural basis for mapping and navigating

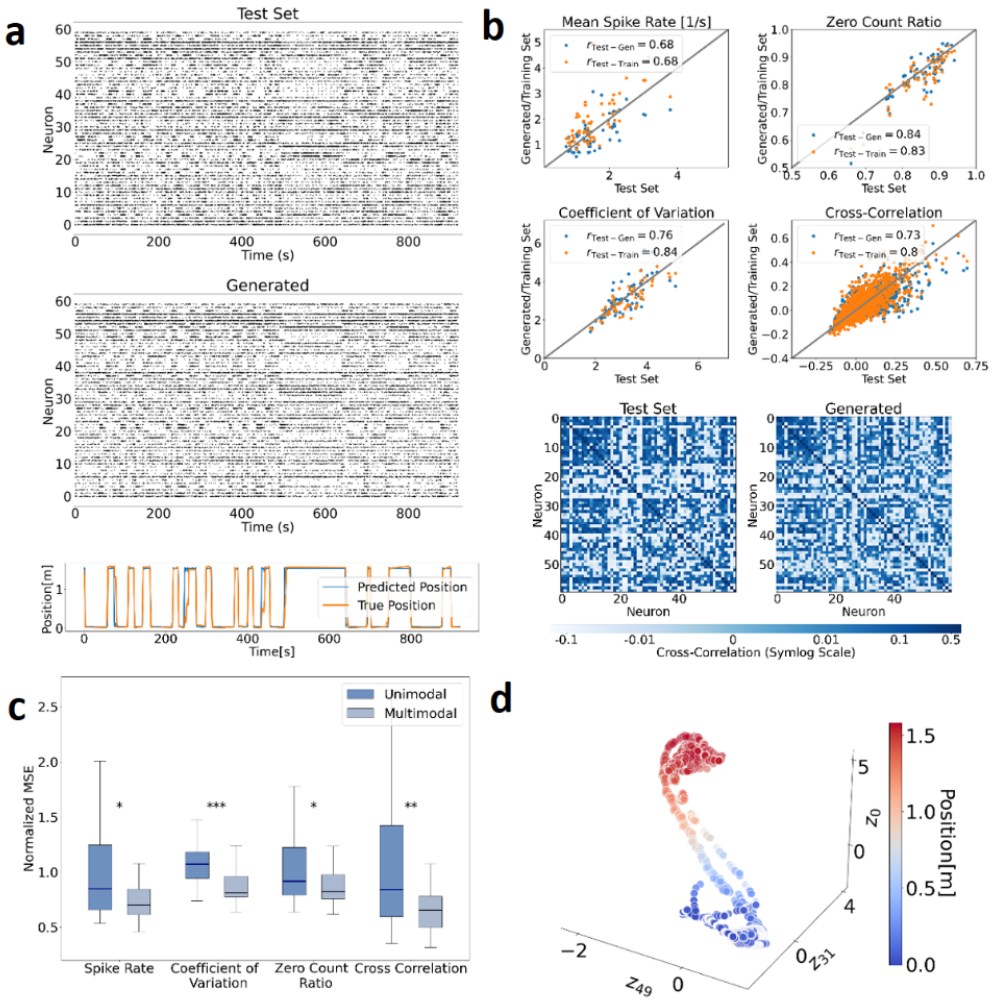

Figure 32: **a**: Example reconstructions of spike trains and spatial location of a rat moving along a vertical track on the unseen test set (second half of the trial), generated from a data-inferred initial condition. **b**: Correlation of mean spike rate, zero count ratio, coefficient of variation, and correlation between cross-correlation coefficients between all 60 reconstructed neurons between test set and model-generated data (blue), and between experimental training and test set data (orange). Diagonal gray lines are bisectrices. Bottom: Cross-correlation matrices among all 60 neurons for the test set (left) and model-generated data (right). **c**: Joint DSR from both spatial and neural data significantly improves reconstructions across all spike statistics, as assessed by computing the average MSE between spike statistics across all neurons. The MSE was normalized for each statistic for better visibility. **d**: Subspace of the DSR model's latent space, illustrating how the latent dynamics are structured according to the animal's spatial position. Based on [53].

physical spaces. Our results demonstrate that the MTF algorithm can leverage spatial information to better reconstruct the neural activity of the place cells. The observation that this improves its ability to capture and predict spike statistics along the unseen test set hence illustrates the functional role of the neurons in that region in navigation, and the MTF's ability to meaningfully link different observed modalities.

4.4 HIERARCHISATION FRAMEWORK

This section summarizes results using the hierarchization approach described in Sect. 3.3.

4.4.1 *Benchmark Evaluation*

I first assessed the ability of the hierarchical inference framework to discover interpretable structure from the Lorenz-63 (Eq. 110) and Rössler system (Eq. 111). To this end, I sampled relatively short time series of length $T = 2000$ for ten different values of $\rho^{(j)} \in \{26 \dots 60\}$, while for the Rössler system, I tuned $c^{(j)} \in \{3.8 \dots 4.8\}$ leading to multiple time series $\boldsymbol{X}^{(j)}$ here representing different 'subjects' or experimental conditions. Changes in these parameters lead to significant changes in dynamics, where e.g. for the Rössler system, the dynamics moved from a limit cycle to the chaotic regime for larger values of c (see Fig. 33, right).

FEATURE VECTORS MAP TO GROUND TRUTH PARAMETER VARIATION As illustrated in Fig. 16, every time series is trained with an individual feature vector $\boldsymbol{l}^{(j)} \in \mathbb{R}^{1 \times N_{\text{feat}}}$, while projections are shared across all subjects. Since only one free parameter is altered in the ground truth systems, the most challenging but most principled situation is one wherein the hierarchical shPLRNN is trained with only one free parameter per subject, $N_{\text{feat}} = 1$. This forces the system to restrict all subject-specific variation to a one-dimensional parameter manifold. After training, this straightforwardly allows us to relate the extracted feature vector with the ground truth values for $\rho^{(j)}$ and $c^{(j)}$. Fig. 33 illustrates that using this approach, the respective features $\boldsymbol{l}^{(j)}$ after training further allow us to almost perfectly predict ground-truth values for ρ and c via linear regression. Since the training algorithm has no knowledge of the ground truth values, it has to infer them indirectly from the training data. Given the functional form of the DSR model (Eq. 70) significantly differs from that of the ground truth systems, which are formulated as low-order polynomials (Equations 110 and 111), this makes the ability of the hierarchical approach to implement a linear relationship surprising.

4.4.2 *Applications to fMRI data*

I then applied the hierarchization framework to the experimental fMRI data previously evaluated in Sect. 4.3.3 using MTF. For simplicity, I focus here on the unimodal case and trained the hierarchical shPLRNN using GTF on time series from 10 subjects with the least amount of irregularities/artifacts. Although we cannot expect a low-dimensional manifold to sufficiently capture individual differences for the more complex fMRI data, I attempted reconstructions with a relatively low-dimensional feature vector of $N_{feat} = 15$. This approach still allowed the model to capture the unique characteristics of the individual time series, as illustrated for several example subjects in Fig. 34**a**. For a left-out sample subject, the reconstruction of the overall dynamics was feasible, even when only fine-tuning the low-dimensional feature vector on unseen data, as

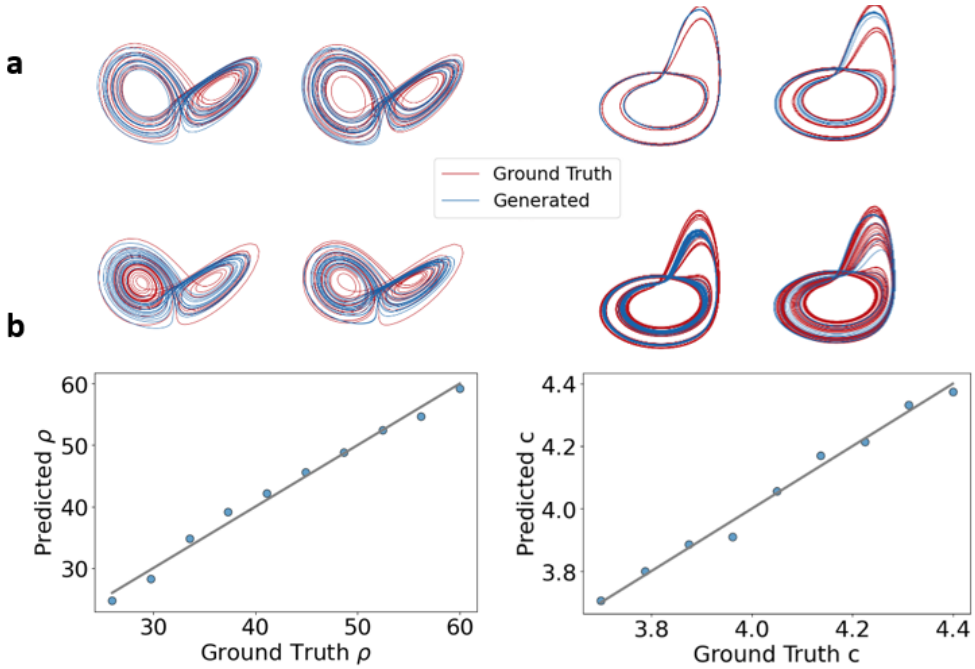

Figure 33: **a**: Reconstructions of the Lorenz-63 attractor (left) and Rössler attractor (right) for different values of ρ and c, respectively, using a hierarchical sh-PLRNN with a one-dimensional feature vector. **b**: By performing linear regression on the one-dimensional feature vectors $\boldsymbol{l}^{(j)}$ after training, the actual ground truth values of $\rho^{(j)}$ and $c^{(j)}$ for each system can be accurately predicted via linear regression.

shown in Fig. 34**b**. This demonstrates the potential of the approach to effectively employ transfer learning across subjects.

Since the feature vectors were highly interpretable for the two benchmark systems in Fig. 33, I investigated whether the identification of the low-dimensional feature vectors is equally robust for the experimental time series. Since the feature vectors are randomly initialized and optimized stochastically, they are not directly comparable across different model instances, yet they should robustly encode differences across subjects. Therefore, I computed a cosine similarity matrix between the extracted feature vectors $\boldsymbol{l}^{(j)}$ for the 10 subjects across 10 different training runs. Subsequently, I calculated the correlation coefficient between the resulting similarity matrices, indicating a strong correlation ($r \approx 0.84$). This indicates a robust extraction of the low-dimensional manifolds, implying that the extracted features could also be employed, for example, for downstream classification tasks such as diagnosing mental illnesses from neuroscientific recordings. Performing unsupervised approaches like clustering on the extracted features can further reveal differences within the observed population. This is illustrated in Fig. 35 for four example subjects from two different clusters determined via k-means on the average cosine similarity matrix, with the extracted clusters aligning with visual differences between the fMRI time series.

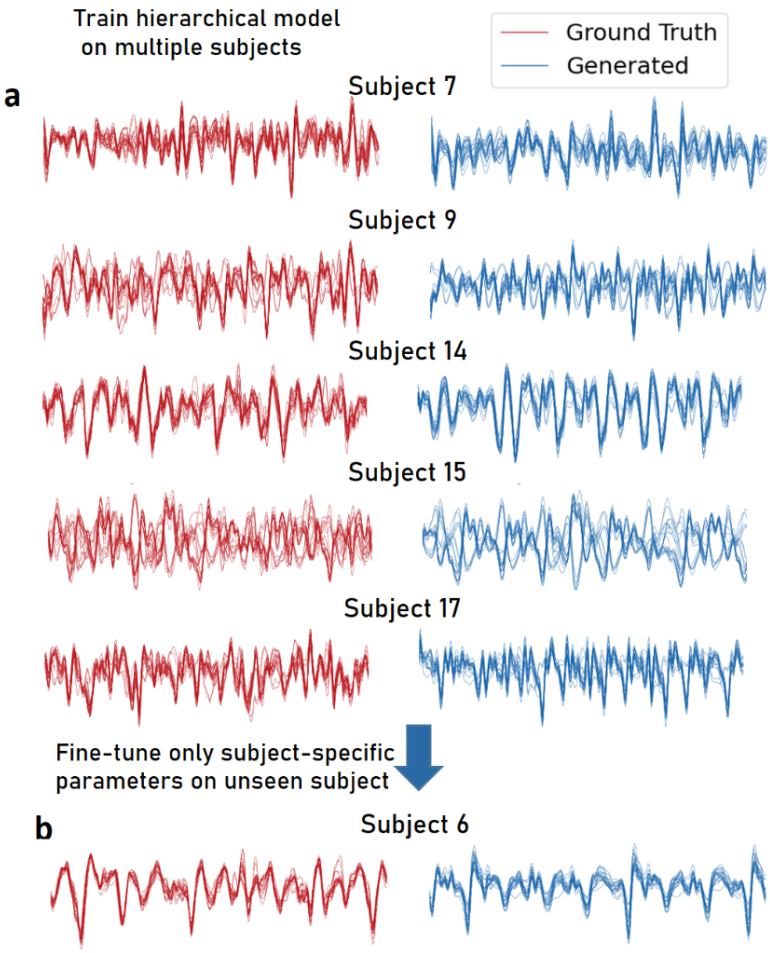

Figure 34: Example reconstructions of several subjects from the experimental fMRI dataset (Sect. A.3.2) using a hierarchical shPLRNN with $N_{feat} = 15$, $L = 300$, and trained using GTF with $\alpha = 0.1$ and $T_{seq} = 72$.

4.5 ANALYSIS OF LINEAR SUBREGIONS OF TRAINED PLRNNS

Piecewise linear activation functions, such as the rectified linear unit (ReLU), decompose complex nonlinear functions into subregions with linear activity [115]. They have both biological and mathematical justifications [141], and have been widely used in ML as activation functions for NNs. The use of piecewise linear functions can significantly simplify the analysis of the complexity of inferred models [283, 362], and can make it easier to understand and visualize how inferred NNs process information. In DS theory, linear dynamics are well-understood and straightforward to analyze, while nonlinear DS lack an equally simple description [58]. This has motivated the modeling of complex dynamics in terms of compositions of locally linear dynamics, e.g. by piecewise linear approximations [64, 77, 169], or by combining linear dynamics with a switching or external forcing mechanism [56, 244, 245].

Consider again the PLRNN defined by Eq. 12:

$$\boldsymbol{z}_t = (\boldsymbol{A} + \boldsymbol{W}\boldsymbol{D}_{\Omega(t-1)})\boldsymbol{z}_{t-1} + \boldsymbol{h} := \boldsymbol{W}_{\Omega(t-1)}\,\boldsymbol{z}_{t-1} + \boldsymbol{h}, \tag{90}$$

To make the dependence of the dynamics on the piecewise nonlinearity explicit, here $\boldsymbol{D}_{\Omega(t-1)} := diag(\boldsymbol{d}_{\Omega(t)})$ is a diagonal matrix and $\boldsymbol{d}_{\Omega(t)} = (d_1, d_2, \cdots, d_M)$

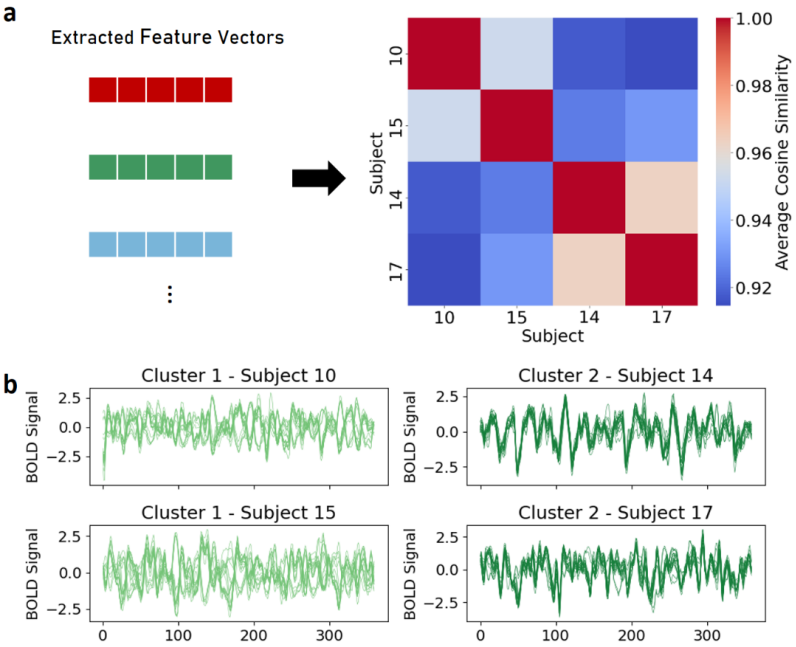

Figure 35: **a**: Cosine similarity matrix based on the average similarity of the feature vectors across ten training runs for four example subjects. **b**: Extracted similarities and cluster labels reflect visual differences in the recorded BOLD signals.

an indicator vector with $d_m(z_{m,t}) = 1$ whenever $z_{m,t} > 0$ and zero otherwise [104].

For the 2^M different configurations of $\boldsymbol{D}_{\Omega(t)}$ as $\boldsymbol{D}_{\Omega^k}$, $k \in \{1, 2, \cdots, 2^M\}$, the phase space of system Eq. 90 is divided into 2^M linear sub-regions with linear dynamics, where

$$\boldsymbol{z}_{t+1} = \boldsymbol{W}_{\Omega^k} \boldsymbol{z}_t + \boldsymbol{h}, \qquad \boldsymbol{W}_{\Omega^k} := \boldsymbol{A} + \boldsymbol{W}\boldsymbol{D}_{\Omega^k}, \tag{91}$$

Using the binary number system, all the sub-regions S_{Ω^k} can be uniquely labeled by an index. A sequence of latent states $\boldsymbol{Z} = \{z_1, \cdots, z_T\}$ can thus be mapped onto a sequence of bitcodes $\boldsymbol{D} = \{\boldsymbol{d}_{\Omega(1)}, \cdots, \boldsymbol{d}_{\Omega(T)}\}$ that encode which linear subregion of the PLRNN the latent state inhabits at each time step.

Assume we have trained a PLRNN model (eq. 90) approximating a DS. In the following, I investigated five benchmark systems: the Lorenz-63 system (Eq. 110), the Lorenz-96 system (Eq. 117), the Rössler system (Eq. 111) and a forced Duffing oscillator (Eq. 120), all in their chaotic regimes, and a bursting neuron model (Eq. 112) implementing a complex limit cycle. All models were trained using STF with a trainable $\boldsymbol{B}$ matrix (Sect. 3.2.3). In the following, the results are illustrated for five example PLRNNs that reconstructed the underlying DS well. However, similar results to the ones presented here could also be robustly reproduced for other trained models.

To assess how the PLRNN model reconstructs these systems in its state space, I sampled a long trajectory $\boldsymbol{Z} = \{z_1, \cdots, z_T\}$ with 100.000 time steps, removed transients of 1000 time steps, and mapped the trajectory onto its bit code representation.

I first investigated the number of unique linear sub-regions traversed by the sampled trajectory for different trajectory lengths, denoted as T. Results for all

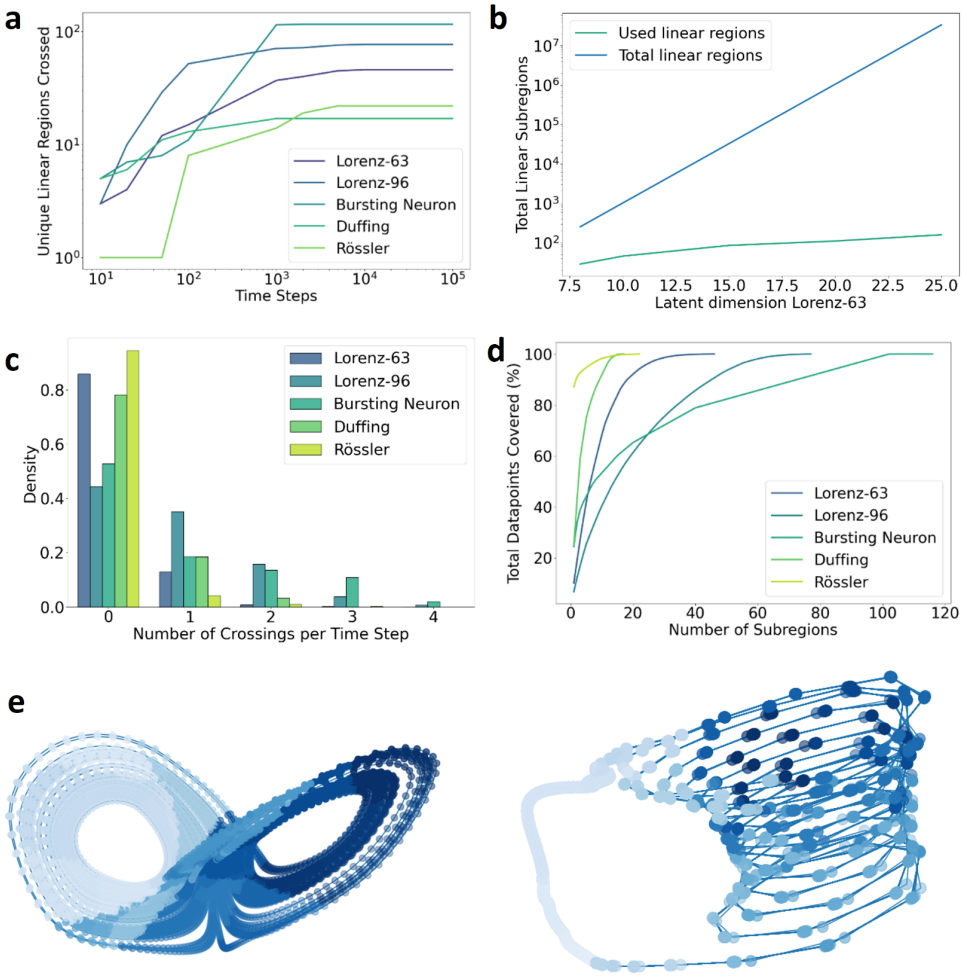

Figure 36: **a**: Plateau effect observed for the number of total subregions traversed for different reconstructed DS and different PLRNN dimensions (Lorenz-63: $M = 10$, Lorenz-96: $M = 30$, bursting neuron: $M = 25$, Duffing: $M = 20$, Rössler: $M = 8$) **b**: Scaling of the total number of linear subregions of a PLRNN, given its latent dimension M, versus the number of subregions inhabited by trained Lorenz-63 systems with different dimensions. While the number of total subregions scales exponentially with M, the used linear subregions increase much more slowly. **c**: Number of boundary crossings per time step for a trajectory with $100,000$ time steps. For the Lorenz-63 and Rössler systems, the models do not cross any boundaries on most time steps, illustrating that the dynamics are highly linearized. **d**: Cumulative frequencies of the individual subregions for trained systems. **e**: Reconstructed attractors for the Lorenz-63 and bursting neuron model, colored with respect to the linear subregions corresponding to each observation.

five benchmark systems are displayed in Fig. 36**a**. For all systems, the total number of unique sub-regions reaches a plateau between approximately $1,000$ and $10,000$ time steps. The plateau level for the total regions is much lower than the total number of available sub-regions for the PLRNNs, given by 2^M for an M-dimensional PLRNN model. This suggests that the reconstructed systems are confined to a substantially lower-dimensional subspace than the total number of linear sub-regions available. To further quantify this plateau effect, Fig. 36**b** compares the total number of linear sub-regions traversed by the PLRNNs with increasing latent dimensions M trained on the chaotic Lorenz-63 system,

compared with the total number of available regions, given by 2^M. Despite the total number of sub-regions increasing exponentially with the model's latent dimension M, the number of utilized sub-regions increases much more slowly.

To assess the extent of local linearity in the dynamics, I analyzed the average number of boundary crossings at each time step. Fig. 36**c** shows histograms normalized over 100,000 time steps. For all systems, zero boundary crossings per time step occur most frequently, indicating that the systems frequently remain within the same linear sub-region for multiple time steps. This result was particularly pronounced for the Rössler system, in which a boundary-crossing occurred on average only every 12 time steps.

The bit code representation $\mathbf{D}$ of the latent sequence can further be analyzed concerning the relative frequency of occurrence of the individual sub-regions in the sequence. Fig. 36**d** illustrates cumulative density plots, comparing the total number of sub-regions required to cover the maximum percentage of data points. Here, the sub-regions were first sorted by their relative frequency. For all models, particularly the four chaotic attractors, a relatively small subset of regions contains most data points, leading to a Pareto-type distribution. This indicates that the system predominantly lives in a much smaller subset $\Omega_n = \{W_{\Omega^{j_1}}, W_{\Omega^{j_2}}, \cdots, W_{\Omega^{j_n}}\}$ of n dominant sub-regions.

Fig. 36**e** illustrates reconstructed systems, with colors corresponding to distinct linear sub-regions mapped onto each observation. Although the mapping from latent space to observation space is not necessarily unique because $M > N$, and hence the kernel of the linear observation model given by $\boldsymbol{B}$ is not empty (for the identity mapping, this can be trivially seen since non-readout neurons do not contribute to the observations). While the observed data points are therefore not inherently associated with specific linear sub-regions, we found that proximal points in observation space were typically related to unique linear sub-regions. To verify this, I conducted proximity matching by defining a threshold distance (e.g. $d = 0.05$, corresponding to 5% of the data variance) and assessing whether proximal points corresponded to different sub-regions. For the Lorenz-63 system, for instance, only 4% of proximal data points within d belonged to different sub-regions, confirming that the attractor is segmented into relatively distinct patches.

Given the set of all linear sub-regions the system traverses, I then examined the average frequency of transitions between all pairs of sub-regions $s, k = S_{\Omega^s} \to S_{\Omega^k}$, sorted according to their frequency (Ω^0 denoting the most frequented sub-region) yielding a matrix of transition frequencies that can be represented as a directed graph between the sub-regions, and further analyzed with graph-theoretical methods. Note that the number of possible transitions naively scales with M^4, and hence this analysis is only meaningful due to the sparse structure of actual transitions in the reconstructed models.

Fig. 37 illustrates these results for a reconstructed Lorenz-63 system. The graph in Fig. 37**a** (top) features a clustering coefficient of 0.31 and high sparsity (only 5 percent of elements in the connectome were non-zero) and indicates small-world structure [418], as assessed by the small world sigma approximated using the `NetworkX` Python package. The visualized graph even mimics the geometry of the true Lorenz-63 system when using the spectral layout in `networkx`. This layout is based on the Laplacian matrix, which is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ where $\boldsymbol{A}$ is the adjacency matrix of the graph, and $\boldsymbol{D}$ is the degree matrix, which is a diagonal matrix where each element is the degree (sum of the weights of

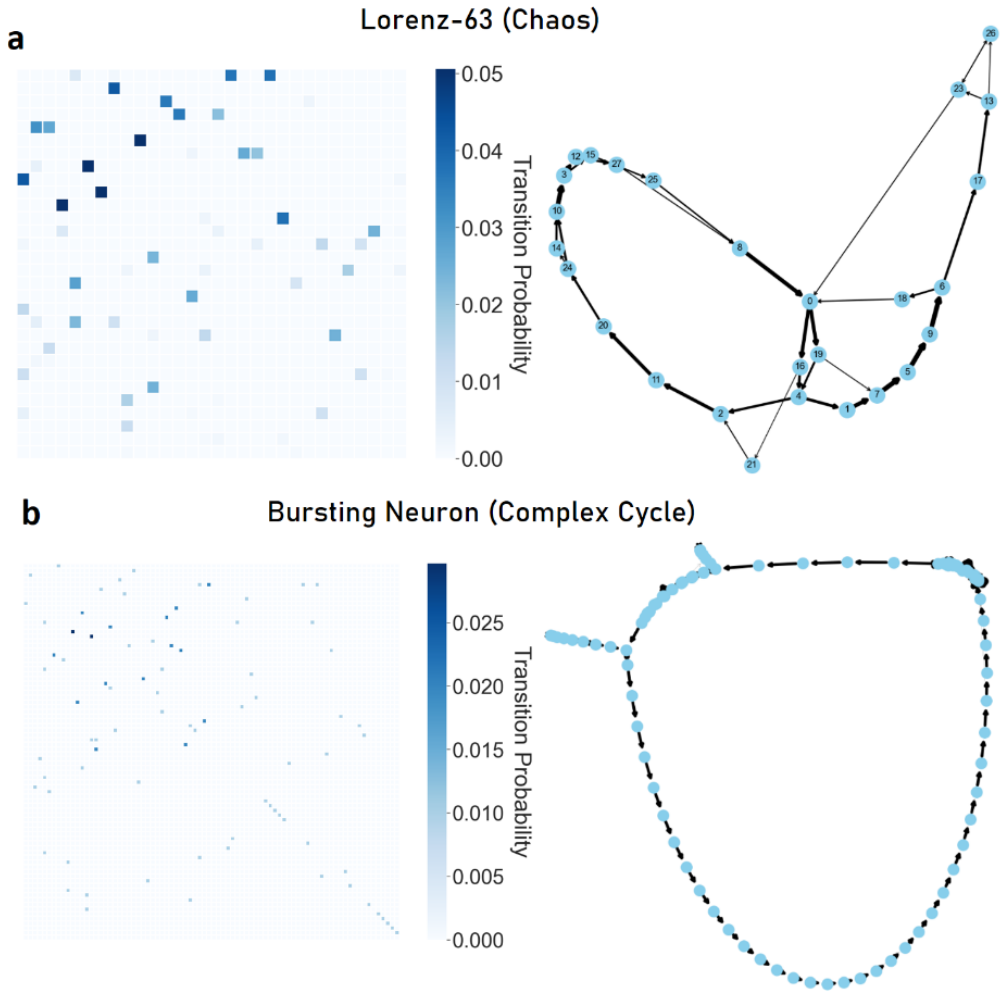

Figure 37: **a**: Connectome of transitions between linear sub-regions, sorted by their relative frequency, for a PLRNN trained on a Lorenz-63 system, and resulting graph structure visualized using the spectral layout in `networkx`. The resulting graph shadows the layout of the real Lorenz-63 system, with the most frequented subregion (label 0) at the center of the intersection between the left and right lobe. **b**: Connectome for a reconstructed bursting neuron model, using the same layout. The graph for this system mimics the cyclic nature of the system and lacks a similarly dominant and interconnected sub-graph.

the edges) of node i. The spectral layout in `networkx` uses the eigenvectors of the Laplacian matrix corresponding to the smallest non-zero eigenvalues as positions for the nodes. This tends to group more tightly connected nodes closer together. The Laplacian is more widely used as a dimensionality reduction technique in ML, for example in Laplacian eigenmaps [31], and has also been used to represent discretizations of PDEs as graphs [372].

The bursting neuron model, being the only non-chaotic benchmark system, but instead implementing a complex cycle, featured a markedly different graph structure than the chaotic benchmarks (Fig. 37**b**). Here, the cycle was implemented in terms of a trajectory passing the same linear sub-regions in the same order for every cycle. Accordingly, most sub-regions were only connected to one other sub-region, with a few exceptions of sub-regions that occur multiple times during the cycle. These regions are connected to several other sub-regions and clustered more strongly in the graph.

The extracted graphs across all four chaotic benchmark systems featured moderate to high levels of modularity $(0.4 - 0.6)$, clustering coefficients $(0.3 - 0.5)$, and small world structure. The presence of small-worldness indicates that while connections between sub-regions are infrequent, the structure of the interactions ensures that any sub-region can be reached from any other through a relatively small number of steps [418]. Overall, these graph theoretic results illustrate how extracted PLRNN models fit locally linear dynamics to the data while retaining a global level of integration necessary to represent the complex state transitions present in chaotic systems. For the bursting neuron model, the clustering coefficient was much smaller (0.07), with the network being highly modular (0.78) and even sparser (1.3%), underscoring that most units are only connected to one other sub-region. These results imply that the obtained graph structure is distinct from the chaotic DS and reflects the non-chaotic nature.

Another way of quantifying the complexity of the connectome is via its entropy, which can be computed for a single node using the Shannon entropy:

$$H_i = -\sum_{j=1}^{M} p_{ij} \log(p_{ij}). \tag{92}$$

Here p_{ij} represents the probability of transitioning from node i to node j, and M is the total number of nodes in the connectome to which node i can transition. Normalizing probabilities p_{ij} for each node ensures that $\sum_{j=1}^{M} p_{ij} = 1$, accommodating for a proper probability distribution over the outgoing connections from node i. The mean entropy H across the connectome can then be averaged over all individual nodes. In line with the previous observations, the entropy for all chaotic systems was much higher (Lorenz-63, $H = 0.51$, Lorenz-96, $H = 0.46$, Rössler, $H = 0.48$, chaotic Duffing, $H = 0.55$) than for the bursting neuron model ($H = 0.11$).

Collectively, these findings show that despite the exponential scaling of the available number of sub-regions for larger PLRNN models, only a small portion of these are used in reconstructed systems. Within this small subset, an even smaller subset of sub-regions contains the majority of the system's dynamics. These sub-regions are interconnected by sparse graphs with small-world characteristics. Since these results generalize across multiple benchmark systems and for RNN models of different sizes, they indicate shared principles in the structural organization of reconstructed PLRNN models. Particularly, reconstructed models do not simply 'overfit' the data but extract low-dimensional and interpretable structures from them that can be leveraged for the interpretation of inferred models. These results also highlight several interesting connections to symbolic dynamics [243, 295]. In symbolic dynamics, the behavior of DS is studied by representing them as symbols and sequences of symbols. A common approach is to divide the state space into a finite number of disjoint subregions, assigning a unique symbol to each subregion, and describing the evolution of the system in time as a sequence of symbols. The collection of all possible infinite sequences of symbols that can be generated by the system's dynamics is called the shift space, and the dynamics of moving between symbols is formally represented by the shift map. This shift map encapsulates the underlying DS in symbolic space and is often of key interest to understand in which way this map represents (and potentially simplifies) the underlying DS. The results from this section can be seen as a specific variant of this approach: the disjoint subregions

are naturally given by the linear subregions of a PLRNN, with each quadrant associated with a symbol, and the shift map is identified with graphs defining transitions between subregions. The results in this section indicate that the graph representation of the shift map could conserve and reflect some important properties of the reconstructed systems (see Fig. 37). These parallels and results indicate that there is much room for further theoretical and empirical investigations.

4.6 OPTIMAL NETWORK TOPOLOGIES FOR DYNAMICAL SYSTEMS RECONSTRUCTION

Here I briefly summarize the approach and key results in [149].

MOTIVATION As discussed in the introduction, training interpretable and parsimonious models is often desirable in the context of scientific ML. However, this approach contrasts with trends in training and employing large-scale foundation models in many ML disciplines. Observations regarding the surprising generalization abilities of highly over-parameterized models, such as the double descent curve [30], further motivate the training of large-scale models. To nevertheless obtain interpretable, smaller models, an increasingly popular approach relies on starting in over-parameterized regimes and pruning out unimportant weights. A well-established and easily applied strategy in ML is magnitude-based parameter pruning, which removes low-magnitude weights in iterative training procedures [42]. The success of these pruning approaches is related to the insight that randomly initialized models often contain 'lottery tickets': these are small subnetworks whose specific random initialization allows them to achieve performance almost equal to that of the full large network [112]. This lottery ticket hypothesis (LTH) has been well-established and investigated both empirically and theoretically across different applications and network architectures [60, 262, 293, 376, 443], such as CNNs in image classification [143]. However, the existence of lottery tickets has not been investigated in the context of DSR.

Real-world DS are often composed of many interacting components, connected through specific network topologies. For example, topologies may arise due to specific physical constraints [176], and the brain consists of firing neurons within a complex network structure featuring scale-free properties, which facilitate global information sharing and local information processing [28, 337]. Consequently, inferring the underlying network topology directly from data has been of long-standing interest [364, 416]. On the other hand, integrating topology with ML approaches has been extensively explored in the context of graph NNs [428]. Given the crucial role of topology in influencing dynamics in real-world systems, this influence has also been studied in DS models such as RNNs and RCs. In RCs (Sect. A.2), the dynamical reservoir is randomly initialized and not trained. Therefore, the expressivity of the reservoir heavily relies on its initialization, such as its topology. Accordingly, numerous studies have investigated the impact of various topologies, such as hub structures, directed graphs, Erdős–Rényi graphs, or those resembling cortical networks, on RC performance [63, 79, 101, 180, 231, 435]. While the influence of topology on RC performance and generalization has been well-documented, these approaches

are not suitable for discovering novel structures that emerge through pruning, as the reservoir weights are not actually trained.

The main contributions in [149] are twofold. First, we demonstrate that the classical approach of magnitude-based pruning is ineffective in the context of DSR, while a pruning method that targets weights based on their influence on state space geometry allowed significant pruning of models. Second, through this pruning procedure, we obtain an interpretable network topology across DSR models trained on different datasets that can be reverse-engineered and used to initialize novel DSR models, leading to faster convergence and the development of more performant, interpretable, and parsimonious models.

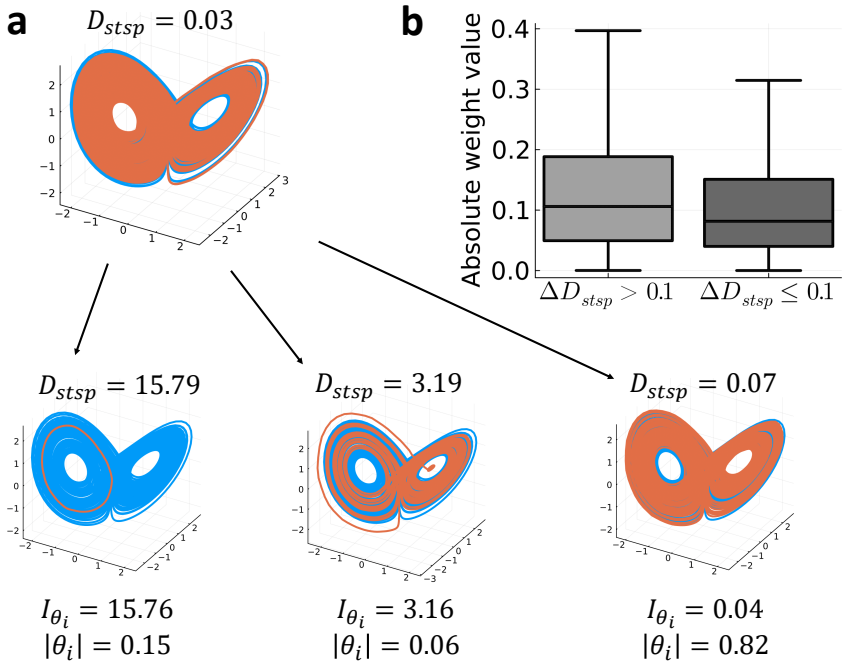

Figure 38: **a**: Illustration of geometry-based pruning. The top row shows a ground truth and reconstructed Lorenz-63 attractor (blue) and a successful reconstruction (red). The bottom row illustrates reconstructions where a single weight was removed with varying influence on attractor geometry. The shift in difference in geometric importance score D_{stsp} does not necessarily relate to the absolute magnitude of the pruned parameter indicated below. **b**: Weight parameters with large ($\Delta D_{stsp} > 0.1$) vs. low ($\Delta D_{stsp} \leqslant 0.1$) impact on geometrical reconstruction quality only feature a small difference in absolute magnitude. This observation illustrates why magnitude can not be meaningfully leveraged for pruning DSR models. Taken from [149]. Created by Christoph Hemmer.

GEOMETRIC PRUNING For the results in [149], we used the standard PLRNN (Eq. 12), trained with id-TF (Sect. 3.2.3), since here the connection between the weight matrix $\boldsymbol{W}$ and network topology is the most straightforward. We use a standard procedure for weight pruning, where pruned weights are given by a mask $\boldsymbol{m}$ that is applied to the weight matrix $\boldsymbol{W}$. The resulting mask $\boldsymbol{m} \in \{0,1\}^{M \times M}$ hence represents the network topology. Traditional pruning methods often assess the importance I_{θ_i} of a parameter by its absolute mag-

nitude. However, in [149] we show that weight magnitude is not significantly linked to performance in DSR (Fig. 38), as measured by geometric agreement D_{stsp} and temporal agreement D_H across several different benchmark datasets already discussed in other parts of this thesis (Lorenz-63, Rössler, bursting neuron, Lorenz-96, ECG, Sect. A.3), and does not outperform random pruning. In geometric pruning, weights that minimally impact the attractor geometry of the reconstructed dynamical system in state space are pruned. This approach results in notable improvements across a range of benchmark datasets and significantly more sparse models.

Computing the geometric pruning measure is more costly than magnitude pruning. It entails generating long trajectories of the reconstructed system and computing D_{stsp} on the respective limit set after removing individual weights, hence naively scaling with network size with M^2. However, since this procedure is independent of each network weight, it can be parallelized. Further, despite the potential downside of increased computational costs, geometric pruning yields interpretable network topologies, enabling the creation of a template for novel initialization.

PRUNED NETWORK TOPOLOGIES LEAD TO IMPROVED DSR The second key result in [149] is that network topology, but not the specific (randomly initialized) weight configuration is essential for the improved performance of the pruned RNNs. The classical LTH [112] states that the combined topology of the 'winning' subnetwork $\boldsymbol{m}$, in conjunction with a specific random initialization of model parameters $\boldsymbol{\theta}_0$ of this subnetwork constitutes the winning ticket particularly well-suited at solving a given task. To test whether this also held for the masks $\boldsymbol{m}$ and weights $\boldsymbol{\theta}_0$ obtained via geometric pruning, we resampled network parameters $\boldsymbol{\theta}_* \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ with a fixed mask $\boldsymbol{m}$ from the same distribution as the initial weights $\boldsymbol{\theta}_0$, and compared this to the scenario where $\boldsymbol{\theta}_0$ is fixed after the initial draw, as in the standard LTH. We found that the network topology, given by the mask $\boldsymbol{m}$, was far more important than the specific weight vector $\boldsymbol{\theta}_0$: Redrawing $\boldsymbol{\theta}_*$ from scratch vs. fixing it to the initial $\boldsymbol{\theta}_0$ did not lead to significant differences in DSR performance across a range of datasets. In the 'classical' LTH, masks and weight distributions are tied to each other in a specific way. While lottery tickets are bound to appear in highly overparameterized networks, they can therefore not be straightforwardly reverse-engineered. However, since in our case, network topology was the determinant factor, we can study the obtained topologies independently of specific weights, and extract invariant structures that can be reverse-engineered to initialize new models. This procedure is illustrated in Fig. 39.

When analyzing the network topologies of models pruned with geometric pruning, we found topologies characterized by both hub-type structures and small-worldness, such as present in the famous Watts-Strogatz model [418]. This means that graphs have a small average path length, as well as a high clustering coefficient. At the same time, the networks feature a hub-like structure with a few highly connected network nodes and many sparsely connected nodes, such as in the Barabási-Albert model [7, 24]. We hence call this topology GeoHub (for geometrically-pruned-hub network). Example graphs for these classical network topologies, compared to the one obtained with GeoHub, are displayed in Fig. 40. Our experiments in [149] show that networks initialized with GeoHub lead to overall best performance in DSR tasks across a range of sys-

tems, followed by the Barabási-Albert model, and also train significantly faster than their counterparts. It is interesting to note the connections between these results and the ones in the previous section 4.5. While here the obtained graphs relate to completely different aspects of the trained models (the connectome of the linear subregions vs. the connectome of the network weight matrices themselves), in both cases, we obtain small-world-type sparse graphs that indicate mechanisms by which PLRNNs approximate complex DS.

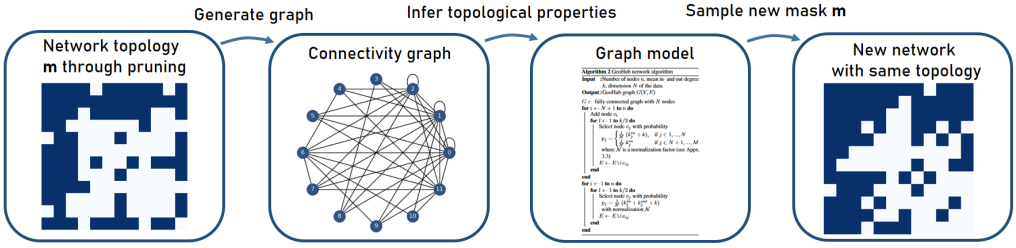

Figure 39: Approach for translating graph-topological properties of trained networks into a general scheme to be used as topological prior. From [149].

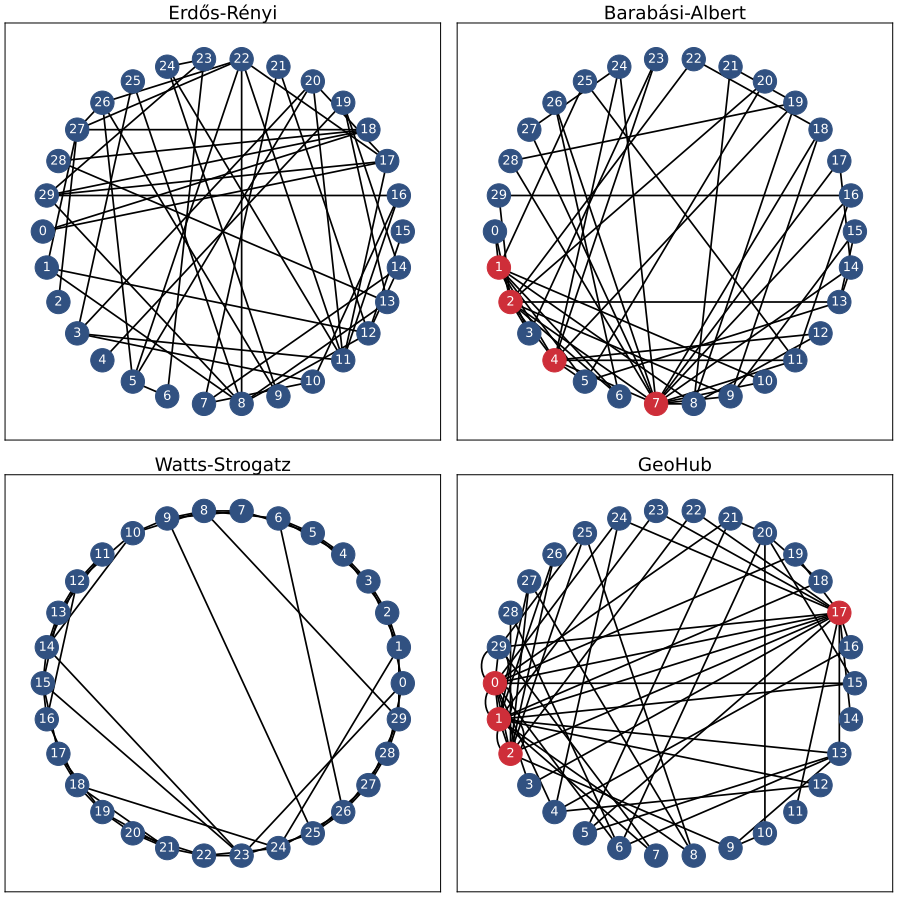

Figure 40: Example graph topologies with network sparsity of 85%. Hubs with $\geqslant 6$ connections are marked in red. From [149].

5

DATA-DRIVEN LEARNING OF SOCIAL INTERACTION DYNAMICS

This chapter outlines the main results from Brenner et al. [55], providing a direct use case of the MTF framework (Sect. 3.2.5) on discrete experimental time series. Since this study constitutes a relatively self-contained project, I included some additional methodological details and a description of the dataset directly within the chapter.

5.1 INTRODUCTION

COMPUTATIONAL MODELING OF SOCIAL INTERACTIONS Social interactions significantly influence emotional well-being and relationships [287, 329]. Successful interactions can reduce loneliness and depression while enhancing positive emotions, self-esteem, and a sense of purpose and meaning in life [326]. However, mental disorders, such as autism spectrum disorder and borderline personality disorder, can impair one's ability to engage in social interactions, often exacerbating other problems that come with these disorders [113]. Understanding and predicting interaction behavior is crucial for developing therapeutic approaches and comprehending the psychological underpinnings of human social behavior.

Social exchange games in laboratory settings offer a structured method to study social interactions and decision-making. The trust game (TG), for example, involves participants in repeated interactions, enabling the analysis of evolving behaviors and decisions [36]. Playing this game successfully entails a variety of cognitive processes like trust evaluation, risk assessment, and emotion regulation, making investment behavior in the TG a complex decision-making task [248]. Computational models have been developed to understand behaviors in the TG, employing process-driven generative modeling approaches, such as reinforcement learning (RL) [67, 163, 164]. These models allow for the dissection of decision-making aspects into interpretable parameters. However, models are usually based on the designers' prior knowledge and tailored to the specific experiment, which can make model building laborious and error-prone [98].

Recently, data-driven models, particularly RNN models, have emerged as alternatives for learning the computational mechanisms underlying behavior from time series data [103, 384, 407]. Since these models are learned in an entirely data-driven manner, they can uncover novel hypotheses about mechanisms underlying social behavior. Despite their potential, RNNs have not been widely applied to study social interactions, possibly due to the often short and noisy nature of experimental data, the lack of models for inferring nonlinear dynamics from ordinal data, and the challenges in analyzing and interpreting RNNs. The MTF framework discussed in Sect. 3.2.5, its promising results in DSR from purely ordinal data, discussed in Sect. 4.3.2, and the mathematical tractability of the introduced DSR models (Sect. 3.1) motivated their application as a data-driven approach to study social interactions. Once inferred, the RNN model serves as a 'digital twin' of the real-world entity [275], optimally

mimicking the complex social decision style of the participant it was trained on. Digital twins have found widespread use in medicine for their potential to yield mechanistic insights into the process of interest beyond the scope of the recorded data (which is often difficult to come by). For instance, digital twins have been used in cardiovascular medicine [75], where digital twins of patients' hearts are created to simulate cardiac function [289], and can aid in developing new interventions such as drug therapies or surgical procedures. Learning digital twins of social interaction dynamics can similarly advance our understanding of complex decision-making processes, and the inferred model can then be used as a virtual interaction partner to simulate new interaction scenarios.

TRUST GAME The behavioral data studied here was collected with 32 students. The students played a TG, where participants acted as investors, engaging with four virtual trustees across multiple rounds, deciding to invest between 10 and 50 monetary units in each round. Their investment is then tripled and given to the trustee, who then returns a portion to the investor based on predetermined ratios (illustrated in Fig. 41**a** and **b**). Participants were randomly assigned to either a social condition, interacting with human faces, or a non-social condition, interacting with geometric shapes, with 16 participants in each group. This assignment aimed to minimize anthropomorphism in the non-social images, and significantly affected investment behavior. Participants received visual input from a trustee, which was characterized by a combination of 'expression' and 'fairness' cues, which determined the average repayment ratio (RR). The fair trustees returned an average of $44.5\% \pm 9.5\%$, and unfair trustees $26.5\% \pm 9.5\%$ of the investment chosen by the participant. In the social condition, expression cues varied across five levels of emotional facial expressions, signaling different repayment ratios, whereas in the non-social condition, the cues were represented by straight lines at different angles on top of geometric shapes. These cues were graded from '++' (highest repayment ratio) to '- -' (lowest), with each '+' or '-' altering the repayment ratio by approximately 7%. Participants did not have any prior knowledge about how cues would relate to repayment ratios and had to infer this during the interaction. Each participant played for $T = 80$ trials, with each trial involving the presentation of a trustee image, an investment decision, and feedback on the outcome. Importantly, the order of conditions, trustee roles, and presence of cues were pseudo-randomized across participants.

MODEL INFERENCE The models were inferred using the MTF framework. As a DSR model we used a dendPLRNN, which is coupled to the same ordered logit ordinal observation model (Sect. 3.2.7, Eq. 51), predicting an investment of $a_t = \{1, 2, 3, 4, 5\}$, at each time point, relating to the five investment of 10 up to 50 (here simply scaled with a factor of 10) monetary units invested per trial. Since the ordinal observation model specifies the probabilities of selecting each of the $K = 5$ choice options at each time point, conditioned on the latent states at that time, $p(a_t = k \mid z_t)$, the choice entropy can be straightforwardly assessed by computing the Shannon entropy of this choice distribution:

$$H(z_t) = -\sum_k p(a_t = k \mid z_t) \log(p(a_t = k \mid z_t)). \tag{93}$$

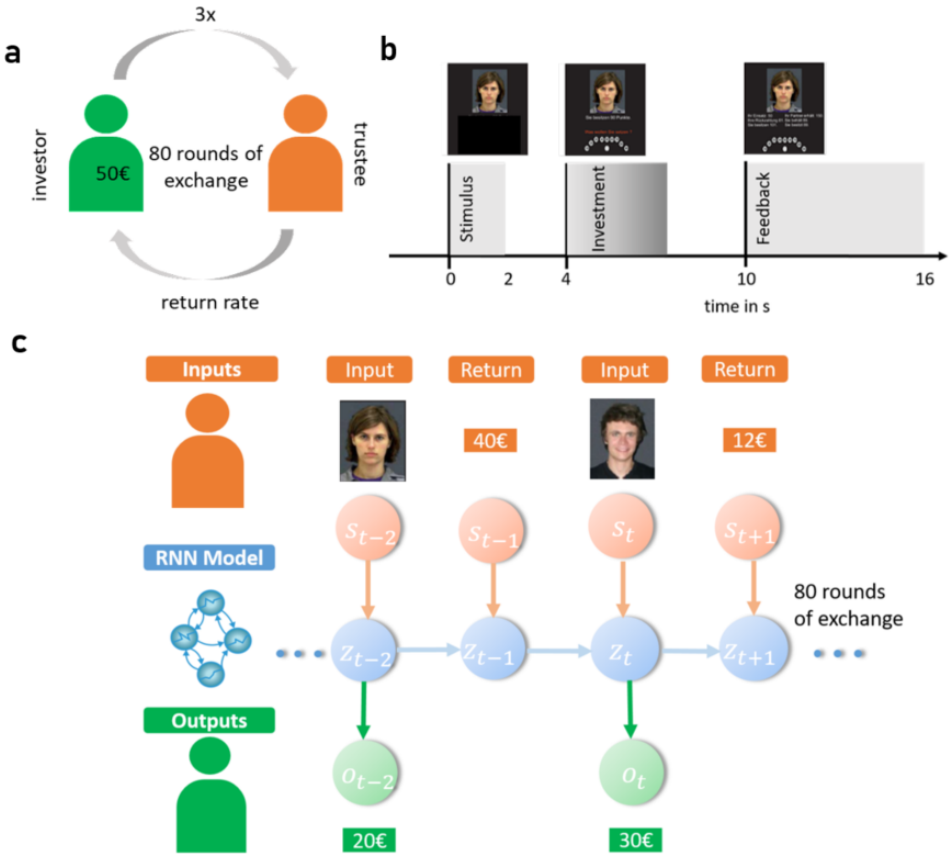

Figure 41: **a**: The Trust Game (TG) setup, where participants, presented with an image of one of four virtual trustees, are given 50 fictitious monetary units for investment. The invested sum is then tripled and passed on to the trustee. Participants subsequently receive feedback detailing their investment, the trustee's repayment, and the retained amount. **b**: Example trial. **c**: The RNN model training process, mimicking the TG setup. The RNN, after receiving fairness and expression inputs, forecasts future investments. Based on its forecasts, the RNN is updated with data on the repayment and new balance.

As the encoder, we used the same temporal CNN encoder (Sect. 3.2.6) also employed for the other results using MTF (Sect. 4.3), but using a causal masking of the time window so future values can not be used to infer current states. The effectiveness of the CNN encoder in learning time delay embeddings from partially observed data (see Fig. 27) makes it a suitable choice here, especially considering that the behavioral investment data constitutes only a partial observation of the complex latent dynamics underlying investment decisions. The dendPLRNN, given as before by

$$z_t = Az_{t-1} + W\left(\sum_{b=1}^{B} \alpha_b \max(0, z_{t-1} - h_b)\right) + h + Cs_t + \epsilon_t, \tag{94}$$

is here trained with external inputs s_t that represent the visual stimuli presented to the participant in each trial of the TG as

$$s_t = (\text{Fair1,Unfair1,Fair2,Unfair2,++,+,0,-,–,Return, Balance}).$$

The RNN is trained in a way that mirrors the participant's experience as closely as possible by splitting the received inputs into two time steps, one for the investment phase and one for the feedback phase, ensuring the model receives the

same visual inputs as participants do in the experiment (Fig. 41**c**). The model then generates its investment decisions during the investment phase.

The input vector includes the four facial identities a participant encounters and the five emotional expressions associated with these identities (entries 5-9), both provided as one-hot encodings. The monetary return from the trustee (entry 10), and the current account balance (entry 11) are provided as numerical values (e.g. 3.2 for a return of 32 monetary units). During the investment phase, the monetary return and balance are initially set to 0, as the participant does not yet have visual feedback on these amounts. At the feedback phase, the actual received return and account balance, are included in the input vector.

Once inferred, the dendPLRNN can be used as a generative model to simulate several different conditions:

- The **true experimental simulation**, where $s_t = s_t$, meaning the simulated input sequence is identical to that presented during the actual experiment.
- A simulation to examine the **effects of trustee and expression inputs in isolation**, where $s_t = c \cdot s_t$ and c is a real-valued tuning parameter.
- Simulations of **'realistic' exchanges** with novel, predetermined interaction strategies of the trustees, in which the model receives updates on return and balance based on its previous investment decision according to some rule.

MODEL VALIDATION Data were divided into a training set (trials $1-60$) and a test set (trials $61-80$), each consisting of two time steps (investment and feedback phase), with models inferred from the training set. To assess the robustness of the inferred models, ten models of equal size were inferred for each participant. Model validation involved evaluating the models' prediction performance using three scores designed to encapsulate different relevant data attributes. First, the mean linear prediction error (MLE) between predicted and actual investments on the test set MLE_{test} and the correlation between predicted and actual investment trajectories on the test set C_{test} (Fig. 42**a**). For the out-of-sample forecasting, models were iterated forward in time from the last training data point (Fig. 42**b**). Due to the potential for incorrect investment predictions that generate misleading feedback (since true returns on the test set reflect responses to the 'ground-truth' investment values), models were re-initialized with accurate test-set investments after every five time steps.

Since from a DSR perspective, we are also interested in capturing long-term statistics in the generative model, which are naturally challenging to model on the very short empirical time series investigated here, we also computed the mean squared error (MSE) between the overall mean of predicted and actual investment responses, grouped by trustee and cue types across all 80 available trials (MSE_{global}) (i.e. difference between blue and orange graphs in Fig. 42**c**). All three scores showed significant correlations with one another, though they assess slightly different aspects ($r_{12} = 0.81, p < .001; r_{13} = 0.32, p < .001; r_{23} = 0.29, p < .001$)). We first normalized the respective metrics across trained models, and computed an aggregate prediction score (PS):

$$PS = \frac{1}{3}\left(C_{test} - MLE_{test} - MSE_{global}\right)$$

Based on this score, we chose a latent state dimensionality of $M = 8$, as this led to a significant overall improvement in the scores when comparing model performance across dimensionalities. Taking $M = 8$, we then selected the model for each participant that had the overall best PS. It should be noted that, in the optimal case, assessing test performance and model selection should be carried out on two different sets (i.e. a validation set and a test set). However, due to the very short available time series, this was not feasible for this dataset. Fig. 42**b** illustrates the observed and forecasted investments for example participants using the highest PS models. Fig. 42**a** compares the mean linear prediction error (MLE) and correlation of these predictions across participants, also including results from a random forecast model. The positive mean correlation (mean $r = 0.16$, $t(31) = 7.8$, $p < 0.001$) indicates a significant out-of-sample predictive performance by the RNN models. Moreover, the models effectively mirrored the aggregate investment behavior of the sample, as depicted in Fig. 42**c**.

5.2 ANALYSIS OF STATE SPACE OF INFERRED MODELS

STATE SPACE ENCODES ENTROPY AND INVESTMENT Beyond reproducing the data, we went on to study the underlying data-generating dynamics. The models predict investments as a probabilistic outcome of the latent RNN states, whose dynamics wholly dictate the generative mechanisms for investment behavior. To elucidate these mechanisms, we analyzed the dominant axes within the state space via principal component analysis (PCA) on each participant's latent spaces. We correlated the first two principal components (PCs) with the investment patterns and the predicted choice entropy (Eq. 93). The first PC showed a significant correlation with entropy (mean absolute correlation $r = 0.36 \pm 0.24$), whereas the second PC correlated with investment magnitude ($r = 0.61 \pm 0.18$; Fig. 43**a** left). These findings suggest that the latent space encodes both the investment decision and its associated certainty. Fig. 43**b** presents two examples of subjects with strong correlations. We call regions in state space associated with high investments and low entropy "highly cooperative" states, while those leading to low investments and low entropy are called "highly non-cooperative" states. Additionally, the predicted choice entropy was demonstrably lower in the social condition ($T(15) = -2.3$, $p = .028$; Fig. 43**a** right panel), indicating that clearer choice preferences might be present given facial cues.

DIRECTIONAL ENCODINGS IN STATE SPACE The input matrix $\boldsymbol{C}$ in Eq. 94 describes the direction and magnitude of displacement in state space induced by each of the 11 inputs (referred to as displacement vectors). To assess the direction of displacement of each cue, we computed average cosine similarities between the first 9 columns of the input matrix, corresponding to the 9 fairness and expression inputs. The cosine similarity measures the orientation of two vectors in high-dimensional spaces relative to each other, normalized by their magnitude, where for instance 1 indicates maximum similarity and 0 indicates no similarity (orthogonal vectors). From these, we constructed a symmetric cosine similarity matrix by averaging across all participants of the social vs. non-social group, respectively, where each off-diagonal matrix element represents the similarity between a pair of inputs. We then employed a

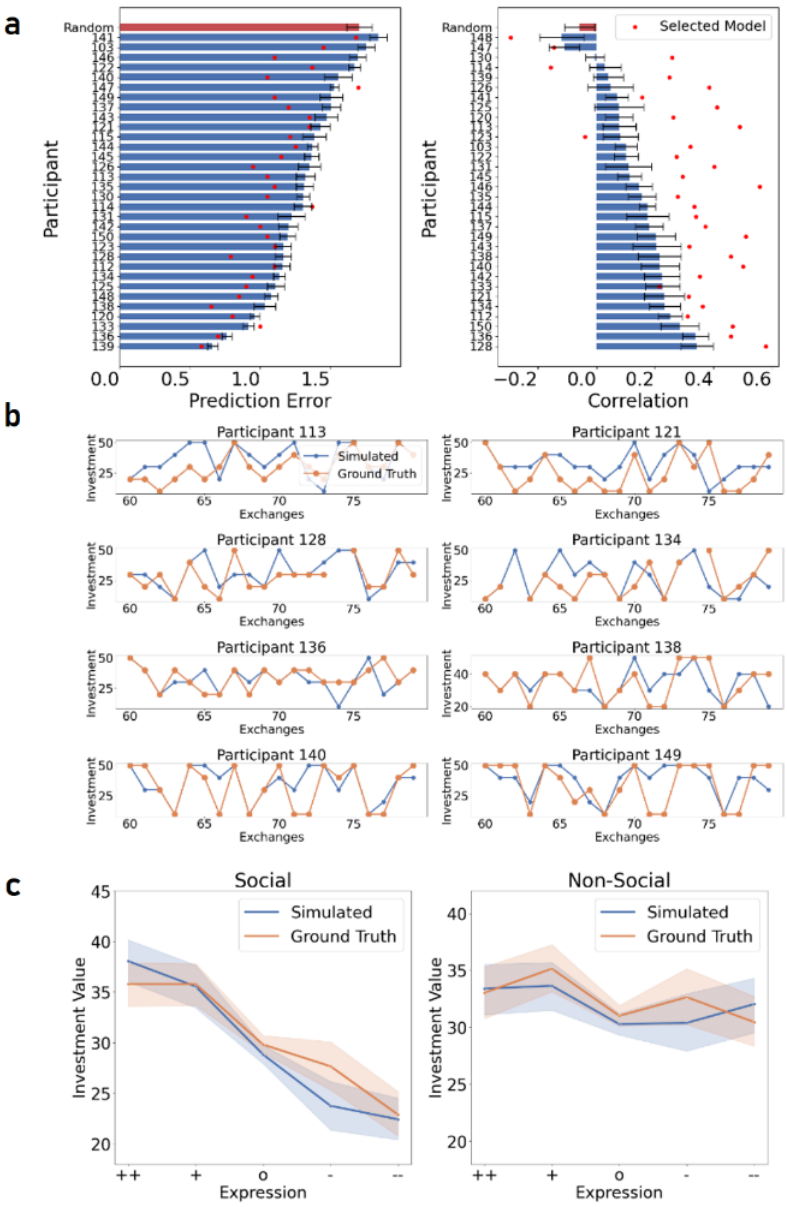

Figure 42: Model Prediction: **a**: Mean linear prediction error (MLE; left) and correlation (right) between predicted and actual investments in the test set across all 32 participants, including a comparison of selected models' performance against random investment choices marked by the red bar. Red dots are selected models, error bars are SEM. **b**: Observed and model-predicted investments for a subset of participants for selected models. **c**: Observed vs. predicted average investment behavior based on the trustee for both social (left) and non-social (right) conditions.

hierarchical agglomerative clustering algorithm on the cosine similarity matrix, using the `sklearn` library in Python with the 'complete' linkage criterion. We could clearly distinguish two clusters and thus two movement directions, one that corresponds to (facial or form) stimulus identity, and one that corresponds to emotional expression (see Fig. 44**a**), which are approximately orthogonal to each other. Mechanistically, these results indicate that participants learned to encode identity and expression independently from one another. For the social condition (Fig. 44**a** top), the individual directions of cues encoded in latent

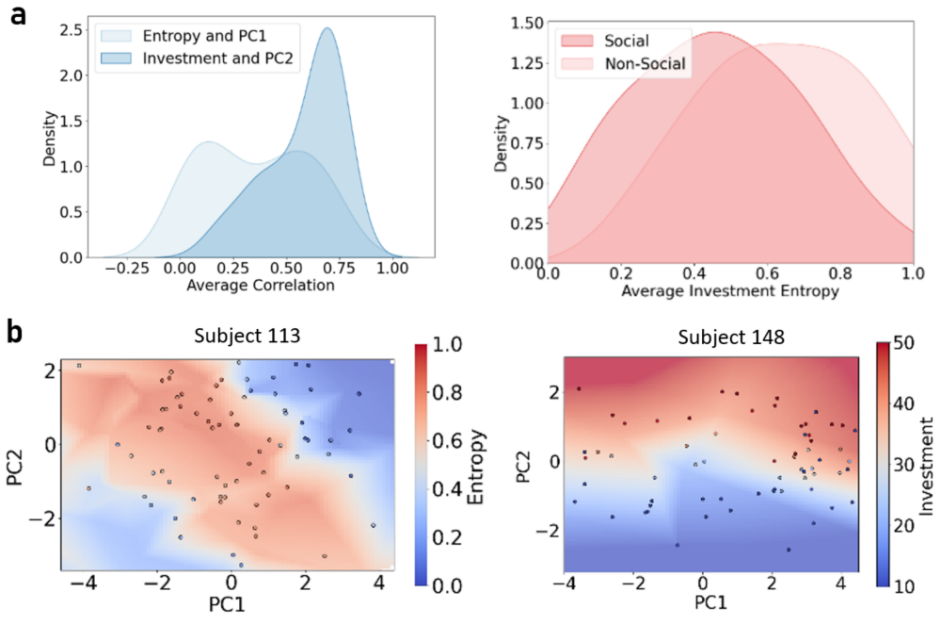

Figure 43: **a** Left: Cumulative density for average correlations between entropy and first principal component (PC; light shade), and investment and second PC (dark shade) across latent trajectories of the true experimental simulation for all 32 participants. Right: Observation model entropy over investments, averaged over actual interactions in social (dark shade) and non-social (light shade) conditions. **b**: State space projection onto the first two PCs for two example participants, structured by an entropy gradient along the first PC (left) and investment gradient along the second PC (right).

space further perfectly reflected the clustering structure into positive, negative, and neutral cues for the expression cues and between the trustees with shifting expression (Fair 1 and Unfair 1) and constant neutral expression (Fair 2 and Unfair 2). Notably, for the non-social condition, the clusters for the expression cues were not correctly identified, in line with the observation that participants struggled to distinguish expression cues in this condition (see also Fig. 42**c**). These results are surprising insofar as the inferred RNNs had no prior notion of what the (binary) input vectors represent, but rather learned the separation entirely from data, i.e. from the investment choices of the participants when responding to the different cues.

Besides examining the direction of displacement in state space caused by an external input, we can also investigate its magnitude. To assess magnitude effects, we compared the Euclidian length of each displacement vector (i.e. columns of $\boldsymbol{C}$) normalized by the respective latent state variance, across inputs. Normalization is necessary since latent dimensions can have different scales across models and subjects. The displacement effects were higher in the social as compared to the non-social condition ($T(15) = 2.08, p = .047$; Fig. 44**b**). Strong expression cues (++ or – vs. + or -) moreover caused a larger displacement than weaker cues for the social condition (paired t-tests: $T(15) = 3.0, p < 0.009$; Fig. 43**b**). While a similar trend could be observed for the non-social condition, the difference between these cues was only marginally significant (paired t-tests: $T(15) = 2.1, p = 0.054$; Fig. 43**b**), consistent with and possibly accounting for the observation of higher entropy in this condition. Collectively, these results indicate that participants are more sensitive to facial expressions as compared

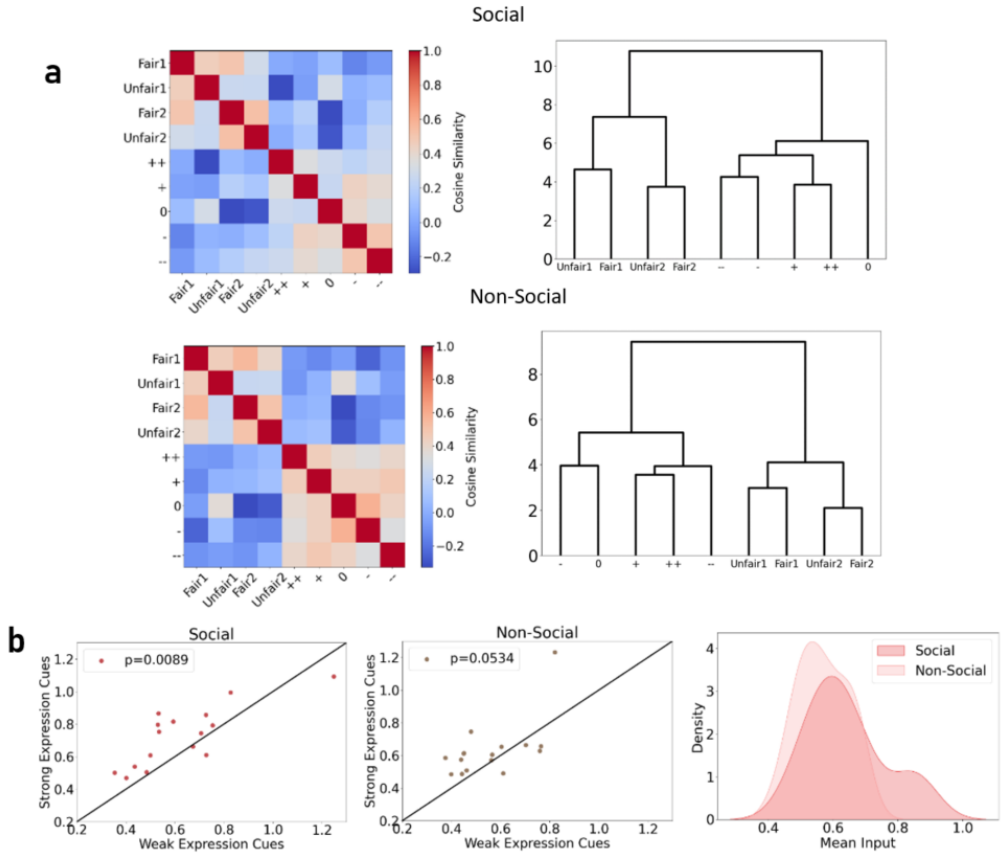

Figure 44: **a**: Cosine similarity matrix of input vectors (left) and resulting clustering dendrogram (right) for social (top) and non-social (bottom) conditions. **b**: Average input strength comparison for strong (++ and –) versus weak (+ and -) expression cues in social and non-social conditions. Bottom right: Average input strength across all cues between social and non-social conditions is higher in the social condition.

to forms, and the models are driven more strongly by strong expression cues than weak ones.

5.3 MODEL SIMULATIONS

One strength of the present approach is that we can use it to simulate behavior in response to different observed and hypothetical inputs. This allows us to perform analyses that go beyond what we can examine in the observed data, gaining additional insight into interaction styles and dynamics, as well as responses to external inputs.

EXTERNAL INPUTS INDUCE BIFURCATIONS IN SYSTEM DYNAMICS We first simulated the effect of inputs in isolation. For these analyses, we presented each input to the inferred models repeatedly over time without repayment and varied the input strength by multiplying the (binary) input vectors s_t with a scaling factor c (e.g. input strength=$c \cdot s_t$, with $c = 0, ..., c_{max}$). These simulations investigate hypothetical scenarios that the models were not trained on (apart from $c = 1$ or $c = 0$), but that could reveal interesting insights into interaction dynamics. For instance, altering the intensity of a 'fair trustee' input can be expected to alter the fairness attribute (e.g., in case of $c > 1$ we increase and for

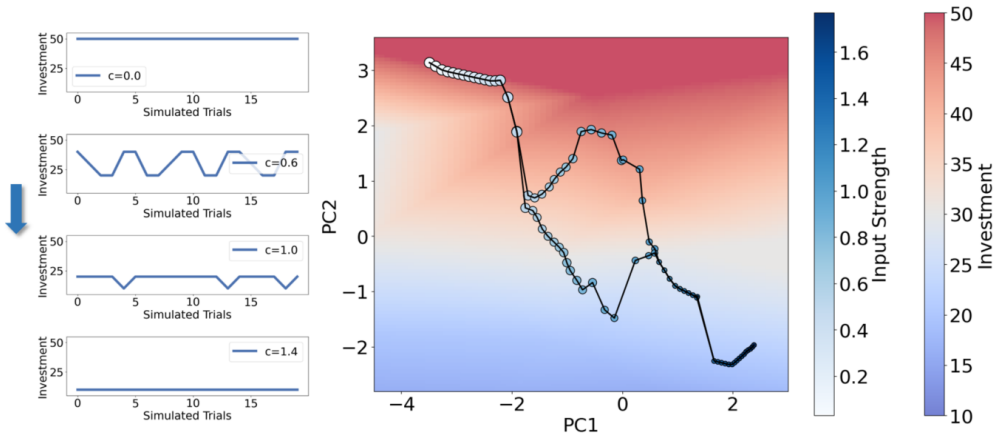

Figure 45: Simulated behavior of an example participant in response to the presentation of an unfair trustee at 4 varying intensity levels ($c = 0, c = 0.6, c = 1.0, c = 1.4$). The investment in the absence of any input is at the maximum investment value of 50. As input intensity is up-regulated, the investment behavior (and corresponding system dynamics) exhibits a qualitative change, also referred to as a bifurcation. The participant first enters an exploratory state ($c = 0.6$), whose precise nature depends on the input strength as well. When the unfair trustee is displayed at full intensity ($c = 1.4$), the investment reaches a minimum (reflecting an unwillingness to cooperate with the unfair trustee).

$c < 1$ we decrease its fairness). Fig. 45 illustrates one such analysis. The input strength is varied with $c = 0...c_{max}$. At 0 intensity ($c = 0$, reflecting no input), the system has an autonomous FP, located in a highly cooperative state. As the intensity c is increased, and with it the intensity of the unfair trustee, the FP location is slowly shifting until a qualitative change in response patterns occurs (a bifurcation; Fig. 45 right). This bifurcation is characterized by the birth of a cycle in which the model switches from maximum investment to now exploring several choice alternatives, in a repetitive fashion. Finally, as the intensity is further increased, a second bifurcation occurs and the system falls back into an FP, now located in the non-cooperative regime. We found qualitatively similar bifurcation patterns for most participants. The RNNs implemented simple FPs for no inputs ($c = 0$) or large inputs ($c = c_{max}$), often located at the minimum (10) or maximum (50) investment value, and exhibited a transition phase characterized by a period of exploration (see Fig. 45). Given that these simulations depict behavior on phenomena not seen during model training, the precise bifurcation onset varied across inferred models of the same participant. However, the general pattern of transitioning between exploitation and exploration was frequently observed in the overall sample. Computing the maximum Lyapunov exponent along trajectories revealed that 26/32 subjects featured a positive maximum Lyapunov exponent for at least some of the cues, while all models also featured negative exponents for other cues, indicating the ability of models to showcase both chaotic/exploratory behavior and stable investment behavior, depending on the presented cues.

SIMULATING DIFFERENT INTERACTION STYLES REVEALS DISTINCT BEHAVIORAL CLUSTERS Finally, we aimed at leveraging the full power of the generative models by simulating end-to-end TG interactions with entirely novel

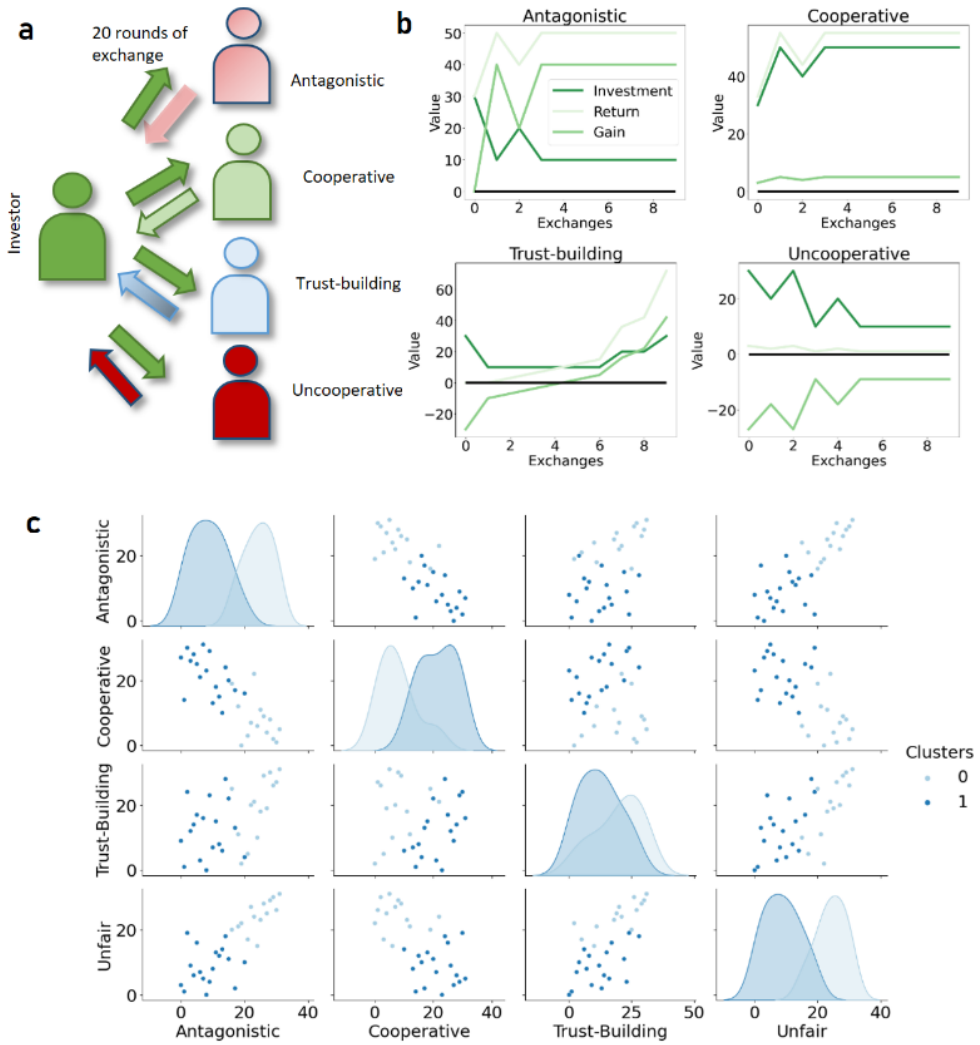

Figure 46: **a**: Interactions of models with four example trustees with different interaction styles. **b**: Example predicted strategies (y-axis) over time (x-axis) for models with successful investment strategies for the antagonistic trustee (AT, the cooperative trustee (CT), the trust-building trustee (BT), and the uncooperative trustee (UT). **c**: Clustering based on average relative gain extracted from different trustees. Low values indicate low ranks, i.e., good performance and high extracted gains. In the off-diagonal panels, the x and y axis denote the ranks of the 32 participants for the respective trustee condition.

simulated trustees with distinct (yet unencountered) return strategies (transferring knowledge of the generative models to out-of-domain interactions; Fig. 46**a**). We simulated four strategically distinct virtual trustees:

- An "antagonistic trustee" (AT) that inversely reciprocates, meaning this trustee exhibits low returns in response to high investments and high returns to low investments.
- A "cooperative trustee" (CT) that consistently reciprocates with an RF of 1.5, returning exactly half of the amount he receives.
- An inconstant "trust-building trustee" (BT) that increasingly reciprocates starting from an initial RF of 0.1 and monotonically increasing to an RF

of 2.0, thus returning less than the invested amount for half the trial and more for the other half.

- An "uncooperative trustee" (UT) that consistently reciprocates with an RF of 0.1.

While these four trustees were selected here because they correspond to intuitive but distinct return strategies, in principle countless strategies could be explored here, tailored to specific research questions. We simulated TGs between the participant-inferred models and these four agents (AT, CT, BT, and UT), recording the average returns each model obtained over a series of 20 interactions. We then ranked the participant models based on the average relative gain over simulated interactions (averaged across all expression and type cues). This gain was defined as the difference between the returned and invested amount, where high values indicate a high gain (since in that case, more is returned than invested), and predicts individual differences in participants' decision-making strategies, and how these may lead to varying levels of success depending on the nature of the person they interact with. The relative gain between AT and CT was negatively correlated (Spearman's ρ=-.72, $p<.001$), and positively correlated between between AT and UT (Spearman's ρ=-.78, $p<.001$) suggesting that models that performed well with the AT performed relatively poorly with the CT, and vice versa. Correlations in performance to the BT were small (ρ=-0.04 with AT, ρ=0.32 with CT, ρ=0.47 with UT), indicating that to obtain a high gain when interacting with the BT may require a different strategy. We then ranked the relative gains of the participant models and applied clustering algorithms (one based on k-means, and a hierarchical clustering algorithm based on the Ward variance minimization) to investigate whether we could identify interpretable clusters with distinguishable interaction styles. Both clustering algorithms lead to similar results. The obtained clusters using k-means are visualized in a pair plot in Fig. 46**b**. We identified two major clusters: a cluster that performed well with the CT, bad with the AT and UT, and moderately to badly with the BT, and a cluster that performed badly with the CT, but well with the AT, UT, and well to moderately with the BT.

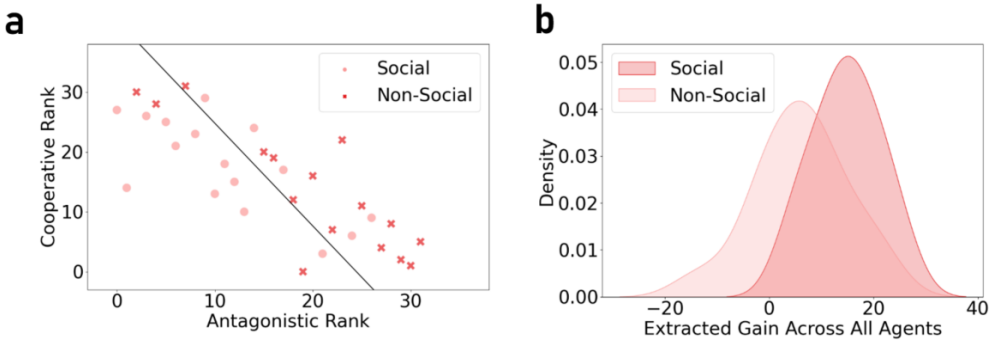

Figure 47: **a**: Comparison of rank for the CT vs. AT for subjects in the social vs. non-social group. **b**: Extracted gain across social vs. non-social group, averaged across all four agents.

IDENTIFIED CLUSTERS RELATE TO SOCIAL AND NON-SOCIAL CONDITION

To understand how the obtained clusters can be interpreted, we trained a linear

classifier to separate the cluster ranks for all four trustees according to the social and non-social conditions. We found the two groups to be relatively well separable even by a linear classifier (Accuracy: 0.81, Precision: 0.81, Recall: 0.81, F1-Score: 0.79, AUC-ROC: 0.86), particularly along the axis separating performance for the AT vs CT (Fig. 47**a**). Notably, most well-performing subjects for the AT were from the social group, while most performing best with the CT were from the non-social group. This was in line with the observation that only 3/16 of the subjects in the social condition belonged to the first cluster, while only 5/16 subjects of the non-social group belonged to the second cluster. The strong relation between obtained clusters and the social and non-social group was further confirmed by a chi-square test of independence ($\chi^2(1) = 6.62, p = 0.012$). Since participants in the first cluster only performed better with the CT, while performing worse with the other trustees, when comparing the absolute gain extracted across all four agents, the social and non-social groups also displayed significant differences, with the social group ($p < 0.0039, t(27) = 3.18$) extracting on average significantly higher gain during the interactions (Fig. 47**b**). The differences between social and non-social conditions indicate that participants learned different interaction strategies depending on the precise nature of the cues that were presented, even though the underlying meaning of the cues did not change across the two conditions, and are in line with the previous observations that models reflect differences between directional encodings for both conditions (Fig. 43**c**).

5.4 INTERPRETATION

Beyond predicting behavioral patterns observed in empirical data, employing the inferred RNNs as generative models in new contexts led to several interesting observations.

First, the dynamics were primarily characterized by movement along axes that encode investment and choice entropy, often partitioning the state space into cooperative (high investment, low entropy), and non-cooperative (low investment, low entropy) regimes. Additionally, participants encoded fairness and expression cues along almost orthogonal directions in the model's latent space, indicating the participant's ability to accurately discern the independent experimental factors (rather than ascribing the effects of cues to specific trustees or vice versa) despite the comparatively short exchange game. Inferring cause and effect and distinguishing between independent causal sources based on environmental interactions is an important human quality. In this game, orthogonal encoding likely facilitates quicker generalization to new situations (e.g., from one smiling trustee to another).

Several additional mechanisms came to light when examining the models for their generative dynamics. For instance, transitions between cooperative and non-cooperative states were often mediated by high entropy regimes characterized by oscillations or chaotic activity. These findings provide evidence that as participants change their strategy from low to high investments and vice versa, they enter an exploratory period in which they initially test different choice options (rather than slowly converging to investments of different order). These findings are consistent with evidence from animal studies that demonstrate that rule-switching behavior is preceded by exploratory hypothesis-building

phases [100]. A clustering analysis of model-predicted investment patterns in response to four strategically different trustees revealed distinct clusters of investment success that aligned well with the social/non-social subgroup. These results indicate that the models might have picked up on subtle differences in interaction styles displayed by the two groups that would otherwise be easily missed from observing the data alone. While validation through future studies is warranted, this simulation approach may help identify distinct interaction styles and elucidate strengths and weaknesses in a player's interaction pattern, possibly beneficial to design novel therapeutic approaches.

The modeling approach based on MTF has several advantages for the analysis of human behavior. As mentioned earlier, data-driven models can be more robust at handling misspecification when compared to process-driven models that rely on (often strong) assumptions about the data-generating process [98]. This makes them flexible in capturing patterns that may not conform to existing theories. This may be particularly relevant to complex decision-making situations, when we lack domain knowledge, or when we assume inter-individual differences are at play that cannot be captured within a common process-driven model. Second, the models can be easily adapted to novel situations or experimental conditions. Due to the mathematically tractable RNN model, we have analytical access to Jacobians and, by that, can easily characterize a system's dynamics (e.g., to differentiate, for instance, whether cue-driven exploratory behavior is chaotic or follows a predictable repetitive pattern). We can further use the model to extensively simulate behavior, both under experimental conditions, and novel, yet unseen settings. This can help to identify and predict distinct behavioral patterns and mechanisms. Many of these analyses cannot be done on the observed behavioral data itself. Finally, since we obtain a generative model of a participant, we could reverse engineer the model to identify settings under which a participant is likely to show cooperative or non-cooperative behavior ([394]). Finally, using the dendPLRNN trained via MTF comes with several benefits. Since the DSR model predicts investments probabilistically, this enables the investigation not only of participants' investment choices but also the (predicted) uncertainty of these choices. Further, the DSR model distinguishes between latent dynamics and observed measurements, enabling the inference of dynamic variables not directly observable from the data. These additional variables can e.g. allow the model to differentiate between subtle differences in the meaning of the cues (Fig. 44). Human social exchanges likely rely on building such latent models and beliefs of the other's intentions, which are frequently only partially observable [114].

6 CONCLUSIONS AND OUTLOOK

6.1 CONCLUSIONS

The methods introduced in this thesis constitute powerful reconstruction algorithms that advance the field of DSR in several directions.

A principal theme connecting the methods and results discussed here is interpretability: the dendPLRNN (Sect. 3.1.1) and shPLRNN (Sect. 3.1.2) constitute DSR models designed to facilitate dynamically interpretable and low-dimensional reconstructions. The hierarchical inference framework (Sect. 3.3.1) aims at extracting low-dimensional interpretable substructure from time series measured across multiple subjects. Sect. 4.5 introduces a general pipeline for analyzing inferred PLRNN models with respect to the linear subregions they inhabit, while Sect. 4.6 introduces a pruning method tailored to DSR that leads to interpretable and parsimonious network topologies. Finally, Sect. 5 discusses applications of the DSR models and training algorithms to experimental psychological data, illustrating ways in which inferred models can be analyzed and simulated to gain insights into complex social behavior.

Another significant focus of this thesis was the application of DSR methods to real-world data. While most of the DSR literature has focused on simple benchmark systems (Sect. 2.5), applying these algorithms to real-world systems to advance scientific insight remains the ultimate goal. Many SOTA algorithms evaluated as comparisons for this thesis struggle with real-world data (see e.g. Fig. 23) due to various challenges, such as noise, partial observations, or non-Gaussian measurements. The methods introduced in this thesis no only achieve good reconstructions from real-world data (Sect. 4.2.2), but even do so from discrete symbolic time series (Sect. 4.3.2) or short multimodal experimental time series (Sect. 4.3.3, Sect. 5). The hierarchization framework (Sect. 4.4) also advances the interpretation of extracted DSR models: it allows transfer learning and data integration across multiple subjects even from short experimental time series.

The MTF (Sect. 3.2.5) approach can be seen as the centerpiece of this thesis. It integrates the tractable RNN models, the SVAE, and the TF-based techniques into a common framework tailored to DSR from real-world data. Its encoder-decoder structure provides a natural way to model (and invert) observation functions common in real-world scientific settings, where missing modalities (Fig. 26), partial observations (Fig. 27), discrete/coarse-grained representations (Fig. 28) or multimodal data (Fig. 32) are commonplace. As such, it can be applied in any discipline where time series observations are measured, including those where the difficulties induced by the measurement process usually impede the application of other DSR approaches, such as those based on symbolic regression [59].

The MTF and hierarchization framework were developed as part of the IMMERSE (Implementing Mobile MEntal Health Recording Strategy for Europe) project. This project aims to promote personalized, patient-centered treatment through the implementation of a Digital Mobile Mental Health (DMMH) tool.

This DMMH tool is used to sample and visualize active survey data collected via the Experience Sampling Method (ESM) and passive sensor data, such as step counts and GPS data, from the everyday lives of service users. As such, the data is naturally multimodal and occurs as both continuous data (e.g. GPS data) and Likert scale ordinal ratings, combined with count observation (e.g. step counts). At the same time, observed time series are often short, contain missing observations, and are recorded from several subjects simultaneously. Beyond this study, these characteristics are the rule rather than the exception in clinical settings, e.g. in psychiatry. One goal of applying ML models in a psychiatric context is the identification of digital 'fingerprints' of mental illnesses. These fingerprints for instance allow for the development of more individualized therapeutic approaches. Models also allow the short-term forecasting of mood trajectories, which can give clues about the onset or offset of acute episodes of mental illness, and which can be combined with external interventions to determine improved outcomes. Some of these applications are already explored within IMMERSE and the AI4U project [153].

In neuroscientific settings, DSR approaches can help elucidate computational mechanisms from neural activity [97]. Changes in neural activity are believed to underlie mental illness [96], so extracting interpretable DSR models can help deliver mechanistic insights into these changes. The advances in neuroimaging techniques like multi-electrode arrays [300] and optogenetics [50] discussed in the introduction underscore the importance of algorithms that perform well on challenging real-world data for progress in computational neuroscience. Particularly the integration of behavioral data and neuroscientific recordings (Sect. 4.3) is an important topic in this context [356], and can help elucidate how the brain integrates information from different channels into its world model [114].

Another significant focus of study in DSR (see Table 1) involves systems of partial differential equations (PDEs), simulating e.g. fluid flows. These systems are usually modeled on a 2D grid, and thus typically have many more state variables (e.g., 128×128) than the benchmark systems evaluated in this thesis. In this context, the development of computationally efficient approximations for these high-dimensional systems, known as reduced order models [10, 235], is of great practical interest, for instance, for effective real-time control [286] of turbulent flows. Although not explicitly evaluated in this thesis, the encoder-decoder architecture of the MTF approach can be extended to this type of data [10]. The encoder model can for example be tailored to spatially extended DS by incorporating two-dimensional convolutions, as commonly used in computer vision. Local interactions between variables common in PDEs could also be incorporated into the DSR models, e.g. by specifying a topological prior as discussed in [149]. As outlined in Sect. 2.7, the search for classical shadows, providing computationally efficient models of complex interacting many-body quantum systems is an area of active research [161], and could leverage the DSR models discussed here.

Reduced-order models are also important in climate science. The earth's climate is a complex DS across many temporal and spatial scales, making it incredibly challenging to model accurately. An important current frontier in climate science is the integration of general ML techniques with pre-existing sophisticated climate models [190], e.g. by leveraging physics-informed ML [321]. The integration of domain knowledge and physical knowledge was also touched upon in several parts of this thesis. In [139], we, for instance, show that out-of-

domain generalization in DS is almost impossible without incorporating strong priors, which are often instrumental in enforcing physically realistic, generalizing models. Hence, leveraging these priors is often essential for applications of DSR models in the physical sciences. Many recent methods build physical priors into ML architectures, for example, by embedding a Hamiltonian structure directly into an RNN [72]. Finding effective ways to integrate priors with the methods introduced here thus remains an important topic for future research.

The results showcasing reconstructions solely from discrete time series (Sect. 4.3.2) also offer much room for future explorations. For instance, language has been likened to a DS before [105]. In language models, language is naturally represented as discrete tokens [342] that encode complex semantic relationships in high-dimensional latent spaces. Reconstructing DS from language data, e.g. reflecting shifts in underlying beliefs and values [398], or even capturing bifurcations in syntactical structures observed during language learning [107], could provide novel and exciting applications.

Beyond this general discussion, in the following, I will specifically address limitations and future research questions for the methods and results discussed in this thesis.

6.2 LIMITATIONS AND OUTLOOK

RNN MODELS While the proposed RNN models (Sect. 3.1) are designed to be mathematically tractable, their formulation remains general, and further constraints can be imposed to aid with interpreting inferred models [97, 212], for instance by enforcing specific network topologies [149].

Another direction to investigate is the integration of trainable and adaptive time scales into the model, similar to LEM networks [341]. Further analysis on how to best apply the manifold attractor regularization ([355], Eq. 20) could also aid in extracting interpretable time scales from DSR models. Combining slow and fast time scales also relates to questions of inferring non-stationary and non-autonomous DS. While non-stationarities are quite common in real-world systems, as already encountered for instance for the hippocampal spike data (Fig. 32) or the social interaction data (Sect. 5), how best to integrate non-stationarities into DSR models is still a topic of ongoing research.

BAYESIAN DATA INTEGRATION The possibility of integrating prior information is of interest in many applications [242]. For instance, in clinical settings, combinations of both time series and other forms of non-temporal data often coexist, and it is not yet clear how best to integrate structural data to improve TS models. Integrating structural information or imposing interpretable structure also relates to the Bayesian framework discussed in Sect. 3.2.2. While a fully Bayesian formulation has proven challenging to train in practice within the SVAE, combining a Bayesian framework with the more powerful, TF-based approaches, or within the hierarchization framework, could lead to novel, more effective ways of integrating priors.

STF AND GTF While the MTF approach was tested in ablation studies both with LSTMs and GRUs (Table 2) and showcased similar performance to the dendPLRNN, how to best utilize approaches like STF and GTF in other model

architectures still requires more research. While GTF is straightforwardly integrated into the MTF framework, MTF has only been thoroughly studied using STF. Using GTF, and benchmarking it against STF on unimodal and multimodal data could therefore be explored further.

The observations linking the smoothing of the loss landscape to the strength of the TF interval τ or TF constant α motivate estimating and adjusting the respective τ and α values during training based on the loss landscape and optimizer directly. For instance, Adam [201] automatically estimates the exponential moving average of the first and second moments of the gradients during training, which could be used as a proxy to provide insights into the local smoothness of the loss landscape during training (Fig. 48). Second order optimizers, such as AdaHessian [433], could provide the curvature estimates of the loss landscape more explicitly, while potentially incurring higher computational cost. Of particular interest would be the curvature of the loss landscape with respect to the TF constant α. Whether these curvature estimates can be computed in a reasonable time, and whether an optimal parameter for α could be deduced and iteratively adjusted from these curvature estimates in a way that outperforms estimates based on the Jacobians outlined in Sect. 3.2.4 remains an exciting topic for further research.

In chaotic DS, future values of the system become decorrelated from past values after characteristic timescales. This property is connected to exponential trajectory divergence and the Lyapunov time [127], which can be used as a criterion to choose the STF interval [276]. It remains an open question under which precise conditions training on a longer sequence length T_{seq} beyond this timescale nevertheless proves beneficial, particularly for chaotic time series, and why and under which conditions TF-based approaches outperform, for instance, truncated BPTT trained solely on short sequences. This discussion about optimal sequence lengths also relates to the question of what role memory plays in DSR. Many sequence models, such as Transformers and structured SSMs, are benchmarked on long-range memory tasks such as the long-range arena [390], and integrating longer sequences, e.g. in transformer-based LLMs, is crucial for enabling understanding across thousands or even millions of tokens [327]. While long-term memory can be integrated into a DSR perspective through slow time scales, e.g. by enforcing manifold attractors [355], it is often not the primary goal. Markovian DS do not contain any explicit dependence on the history of the time series, but future states only depend on the current state (which can still encode all relevant information from the past). Particularly for chaotic DS, memory effectively gets wiped out due to trajectory divergence [127]. While in principle, the training algorithms discussed here can be applied to gated architectures (Table 2), in [149], we observed that applying geometric pruning on an LSTM when training on a chaotic DS led to the pruning of the gating mechanism implementing the long-term memory [155], effectively resulting in a vanilla RNN without gating. Investigating these results further might further help elucidate the role of memory and gating mechanisms in DSR.

MTF Beyond the ablation studies (Fig. 15) already performed, indicating that the combination of losses is crucial for successful reconstructions, optimal weighting schemes for the different loss terms making up the total loss [21] may further improve performance. Another interesting question is to what extent a probabilistic setting benefits the TF approaches: the TF signal in MTF is drawn

from the approximate posterior, and the MVAE is optimized via the ELBO. However, in principle, the TF signal can also be modeled deterministically, for instance by a deterministic autoencoder (AE). On the other hand, in the deterministic TF approaches (STF and GTF), we have often found it beneficial to add small amounts of noise to the TF signal. Work by Ghosh et al. [121] has shown that injecting some noise into deterministic AEs can often provide similar benefits as VAEs while being easier to train. In many ML contexts, including stochastic components during optimization, such as dropout [377], stochastic gradient descent [338] or the inclusion of data/label noise [80], often improves learning and generalization. Trading off stochastic and deterministic components within the MTF optimally thus offers room for further exploration.

HIERARCHICAL INFERENCE FRAMEWORK While the reported results underscore the efficacy of the hierarchization framework, there is still significant scope for future work. An interesting question is whether hierarchically inferred models provide improved reconstructions compared to individual-level models. A second important direction is comparing how extracted feature vectors can be related to other data, such as class labels or survey data for experimental datasets in which this data exists, or to what extent this information can be integrated as a model prior. More generally, the hierarchical framework provides a mechanism for transfer learning in DSR. Foundation models have become popular in many areas of ML, such as large language models, which are trained on a wide range of tasks and can be fine-tuned to specific downstream tasks [160, 292, 448]. While several such models have recently been proposed for TSF [277, 325, 434], how best to design these in the context of DSR, retaining the tractable structure of the PLRNN models while leveraging the power of training on large amounts of data is still an open question.

LINEAR SUBREGIONS ANALYSIS The results presented here allow for several interesting follow-up questions. Particularly, applying this analysis pipeline to real-world systems could provide a new angle through which to investigate inferred models. One could further investigate to what extent new training objectives can be designed to enforce interpretable linearized dynamics. Another direction to investigate is to compare the representations discovered by the PLRNN to other approaches explicitly encouraging the learning of linearized representations of nonlinear DS [56, 77, 244]. Finally, extending this analysis pipeline to other PLRNN models, such as the dendPLRNN and the shPLRNN, in which the relationships between linear sub-regions and reconstructed systems are more complicated, could provide new avenues through which to analyze these models.

SYMBOLIC DYNAMICS The field of symbolic dynamics studies sequences of symbols to understand the behavior of complex systems. Symbolic dynamics were touched upon in two different contexts: MTF allows the reconstruction of chaotic DS from purely symbolic representations (Sec. 4.3.2). Conversely, the results in Sec. 4.5 symbolically represent reconstructed PLRNNs by interpreting the connectome of the linear subregions of a reconstructed PLRNN as a variant of the symbolic shift map.

DSR from symbolic and other discrete representations of DS has hardly been investigated in the literature before. The success of the MTF framework in en-

abling these reconstructions was somewhat unexpected since ordinal or symbolic codings of the dynamics seem to remove much geometric and topological information from the underlying DS. Further investigating how symbolic encodings of DS preserve topological characteristics such as invariants or Lyapunov exponents [162, 268, 294, 442], and the extent to which the original state space topology is retained in DSR from these representations, opens many avenues for future experimental and theoretical research. This includes examining how delay embeddings, already formulated for non-continuous signals in Sauer [347, 348], relate to and can aid in DSR from symbolic representations, which we found to hold empirically for the results in Sect. 4.3.2.

NETWORK TOPOLOGIES AND PRUNING The discovery of an effective network topology for DSR poses several interesting follow-up questions. In recurrent models such as RNNs, where parameters are used repeatedly to generate sequences, the significance of network weight magnitude as an indicator of importance may be less apparent compared to feed-forward NNs, where the LTH has primarily been explored. This leads to the question of whether the effectiveness of the GeoHub topology could be applicable beyond DSR in other time series applications, or whether the resulting network topology is distinct to DSR. Considering the inherent topological characteristics observed in real-world systems, such as scale-free networks or small-worldness, it would be interesting to investigate whether the optimal RNN configuration for a DS could reflect the empirical topology of the underlying DS. Another aspect to consider is whether more efficient pruning criteria, for instance through dimension-wise and parallelizable proxies of D_{stsp} or through other quantities encoding the influence of individual parameters on dynamics (such as the system's Lyapunov spectrum), can be found.

SOCIAL LEARNING The dataset investigated in Chapter 5 poses several challenges. Firstly, the experiment featured 20 input combinations (4 trustees x 5 expressions) but only 60 training data points. Despite the MTF framework performing well even on short time series, accurately inferring DS models on such short sequences remains a challenge, and leads to relatively high variance in outcomes. Combining the training algorithm with the hierarchical inference framework (Sect. 3.3.1) could alleviate some of these challenges by integrating information across subjects. Further, in the repeated rounds of the TG, participants acquired knowledge of the trustee's behavior over time. The RNN model did not explicitly integrate temporal dependencies to capture the learning-induced changes but had to implement them via shifts within its latent space, which however made it challenging to re-initialize the network in a way that accounted for this learning effect. Follow-up studies could integrate a non-stationarity component explicitly, e.g. by providing an external input encoding the current trial number. Finally, the input sequence of the cues was randomized, such that the current investment should not depend on the RF of the preceding trial, and the balance of the participants was reset at every time step. Both of these facts did not encourage the development of longer-term investment strategies by the participants, which would e.g. form in repeated interactions with the same subject, and complicate the extraction of long-term behavioral patterns.

One key advantage of the generic modeling approach is that applications to novel datasets do not require much fine-tuning or adjustments of the algorithm, and can be applied 'off the shelf'. Hence future studies using this approach on other discrete behavioral datasets are in principle straightforward.

Part IV

APPENDIX

A

APPENDIX METHODS

A.1 TRAINING DETAILS

LAYER NORMALIZATION Layer normalization normalizes the inputs to each layer of a NN, and has been widely used to improve RNN training [19]. In [54] we adapted this approach to the PLRNN, often leading to improved performance. To this end, we mean-centered latent states according to:

$$z_t = Az_{t-1} + W\phi\big(\mathcal{M}(z_{t-1})\big) + h_0, \tag{95}$$

where $\mathcal{M}(z_{t-1}) = z_{t-1} - \mu_{t-1}$, which can be be rewritten as a linear matrix-multiplication

$$\begin{aligned}\mathcal{M}(z_{t-1}) &= z_{t-1} - \mu_{t-1} \\ &= \frac{1}{M}\begin{pmatrix} M-1 & -1 & \cdots & -1 \\ -1 & M-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & M-1 \end{pmatrix} z_{t-1} = M z_{t-1}.\end{aligned} \tag{96}$$

The linearity of this operation implies that all theoretical results about the RNN models still hold [54].

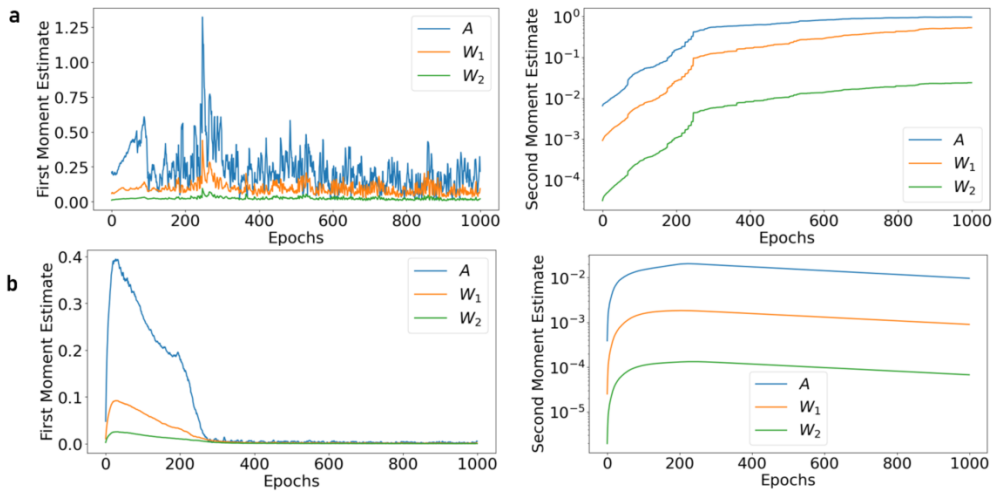

Figure 48: Estimates of the first and second moment from the Adam optimizer for different parameters of the shPLRNN during training on the chaotic Lorenz-63 system. **a**: Without TF, even for a moderate sequence length (here T = 30), gradients retain high variance, making successful training impossible. **b**: With GTF ($\alpha = 0.2$), gradient moments decrease over time after a short initial period with high variance, stabilizing training even for much longer sequences (here T = 200).

OPTIMIZATION In all experiments, we used Adam [201] and its variants, such as Radam [249], as optimizers. We further trained with an iterative learning rate schedule, starting with a learning rate of 10^{-3}, which exponentially decayed to 10^{-6} during training. Using an adaptive optimizer led to significant improvements over naive stochastic gradient descent (SGD). Adam estimates the first and second moments of the gradients, using a moving average model to adjust the learning rate individually for each parameter. To illustrate the importance of this approach, Fig. 48 displays the first and second moment estimates for different parameters during the training of a shPLRNN using GTF. There is a distinct hierarchy in the average estimates for the different parameters. The $\boldsymbol{A}$ matrix features much higher values, followed by the $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ matrices. The values in the $\boldsymbol{A}$ matrix encode the time constants of the states of the shPLRNN, which have a direct and strong influence on the resulting long-term dynamics of the DSR model (see also Fig. 13). This results in a larger variance of the gradients with respect to these parameters. The marked difference between the moments of the gradients at the beginning versus later stages of training also illustrates why learning rate schedulers and the annealing protocol that adjusts α during training, as discussed in [152], often lead to better outcomes.

IMPORTANT HYPERPARAMETER SETTINGS In all papers, we performed grid searches over important hyperparameters for our methods and the comparison comparison methods. As already noted in other parts of this thesis, for training with STF and GTF, τ and α had the largest impact on performance, so we performed line searches for $\tau \in \{1, 5, 7, 10, 15, 20, 25, 50\}$ and $\alpha \in [0, 1]$ in steps of 0.05. For the latent dimension M and basis expansion B of the dendPLRNN, and the hidden size L of the shPLRNN, we observed that increasing the model size generally enhanced performance up to a certain threshold (also see Fig. 8). Therefore, we selected model sizes around the point where this threshold effect set in. The parameters of the dendPLRNN $\boldsymbol{A}$, $\boldsymbol{W}$ and $\boldsymbol{h}$ were initialized according to [389], and drawn from a uniform distribution for the rest, where the initial ranges of the distribution of the thresholds $\{\boldsymbol{h}_b\}$ was determined by the extent of the data [54]. For the shPLRNN, we used a uniform distribution for all parameters [130].

A.2 COMPARISON METHODS

A.2.1 *Unimodal Comparisons*

There is a plethora of approaches for data-driven DSR. This section gives an overview of several techniques from the most important classes of DSR algorithms, all of which have been employed as comparisons in Chapter 4.

SINDY This summary closely follows the one I wrote for [139], based on the introduction of the algorithm in [59]. The Sparse Identification of Nonlinear Dynamics (SINDy) algorithm aims at deriving a sparse representation of governing dynamical equations from observed data. Consider a set of measurements $\mathbf{x}(t) \in \mathbb{R}^n$, where n represents the count of system variables and $t = t_1 \dots t_m$ denotes the times of observation. The first step in applying SINDy involves

numerically approximating the flow $\frac{dx}{dt} = \dot{x}$, typically using finite difference methods. As per [59], these derivatives are structured into a matrix form:

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^\top(t_1) \\ \dot{\mathbf{x}}^\top(t_2) \\ \vdots \\ \dot{\mathbf{x}}^\top(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix} \tag{97}$$

In the subsequent optimization phase of SINDy, the objective is to identify a sparse matrix of regression coefficients $\boldsymbol{\Xi}$ satisfying:

$$\dot{\mathbf{X}} = \Theta(\mathbf{x})\boldsymbol{\Xi}, \tag{98}$$

where $\Theta(\mathbf{x})$ denotes a pre-defined library of candidate functions applied to the state variables $\mathbf{x}$, exemplified as:

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^2 & \mathbf{X}^3 & \mathbf{X}^4 & \cos(\mathbf{X}) & \dots \\ | & | & | & | & | & | & \end{bmatrix}, \tag{99}$$

To determine the regression coefficients, a sparsity-promoting optimization method, such as LASSO regression or the Sequentially Thresholded Least Squares (STLSQ) algorithm, is used to solve for $\boldsymbol{\Xi}$. For all the experiments relating to SINDy, we used the Python implementation (`PySINDy`, [368]).

Formally, SINDy defines a class of finite-dimensional linearly parameterized functions with m differentiable basis functions $\psi_i : \mathbb{R}^n \to \mathbb{R}, i = 1, ..., m$,

$$\mathcal{B}_L = \left\{ f_j(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{i=1}^{m} \theta_{i,j}\psi_i(x) \,\middle|\, \forall j, \boldsymbol{\theta} \in \mathbb{R}^{m \times n} \right\}. \tag{100}$$

In [139], we show that SINDy can reconstruct any (potentially multistable) DS from a single trajectory $\Gamma_{x_0} \subset \mathcal{D}$ provided it is not solving an algebraic equation in the parameters (Eq. 100), and given the correct set of library functions. In this case, the OODG problem is strictly learnable (Eq. 89). These observations are illustrated in Fig. 49 for the following two-dimensional ODE system:

$$\dot{x} = x + x(x^2 + y^2 - 1)(4x^2 - 4xy + 4y^2) + (x^2 + y^2)(-2x + 2y + x^3 + xy^2),$$
$$\dot{y} = y + y(x^2 + y^2 - 1)(4x^2 - 4xy + 4y^2) + (x^2 + y^2)(-2x - 2y + y^3 + x^2y).$$

This system has one stable (inner cycle) and one unstable cycle solution (outer cycle). The unstable outer cycle solves an algebraic equation, leading to an incorrect reconstruction. On the other hand, SINDy correctly infers the vector field from the inner trajectory.

We also considered another VF that has the same solution solving an algebraic curve (the unit circle), but is composed of both polynomial and trigonometric functions:

$$\dot{x} = 2y\cos(x),$$
$$\dot{y} = x^2\sin(x) - 2x\cos(x) + y^2\sin(x) - \sin(x).$$

Results for different function libraries and observed trajectories are illustrated in Fig. 50. SINDy also excels at reconstructing other multistable DS, such as the multistable Duffing system, given the correct library is provided, but fails to generalize and only locally approximates dynamics if this is not the case (Fig. 51).

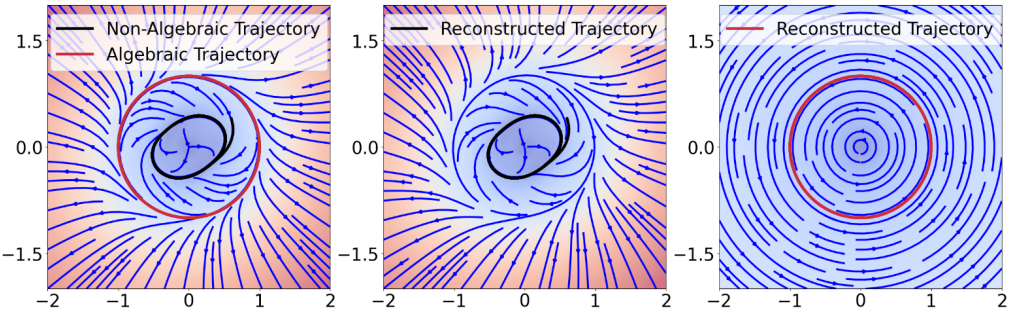

Figure 49: Example reconstructions using SINDy, comparing reconstructions from a trajectory solving (red) and not solving (black) an algebraic equation in the parameters. From [139].

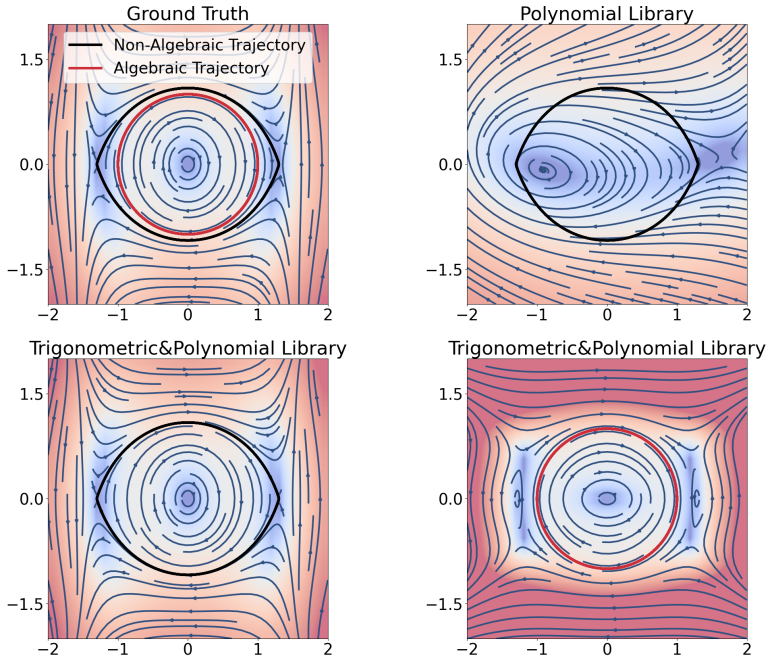

Figure 50: For the GT shown top-left, providing SINDy only with a polynomial library leads to an incorrect solution for the VF (top-right), while providing the correct library including both trigonometric and polynomial functions (and mixing terms) leads to the correctly inferred VF (bottom-left). Finally, even with the correct library, providing as data the curve solving an algebraic trajectory leads to an incorrectly inferred VF (bottom-right). From [139].

RESERVOIR COMPUTERS Reservoir Computing (RC), first suggested in [37, 174], is a general framework for training RNN models which avoids the exploding- and vanishing gradient problems discussed in Sect. 3.2.1. The central concept in RCs is the use of a large, fixed, and random RNN, known as the reservoir. The dynamics of the reservoir can be described by:

$$\mathbf{r}_{t+1} = \Phi(\mathbf{W}_{in}\mathbf{u}_{t+1} + \mathbf{W}\mathbf{r}_t) \tag{101}$$

Here, $\mathbf{r}_t$ represents the state of the reservoir at time t, $\mathbf{u}_t$ is the input at time t, $\mathbf{W}_{in}$ is the weight matrix for input connections, $\mathbf{W}$ is the weight matrix for connections within the reservoir, and Φ denotes a nonlinear activation function, often the hyperbolic tangent function, applied element-wise. The general procedure is illustrated in Fig. 52.

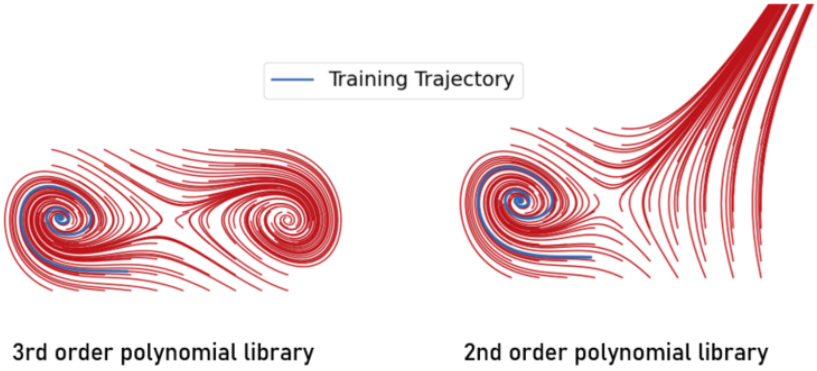

Figure 51: SINDy needs the proper function library to correctly infer a system across the whole state space (left). If the 3rd order term present in the Duffing equations is lacking (right), the inferred VF may only be locally correct (or not at all for more complex systems). From [139].

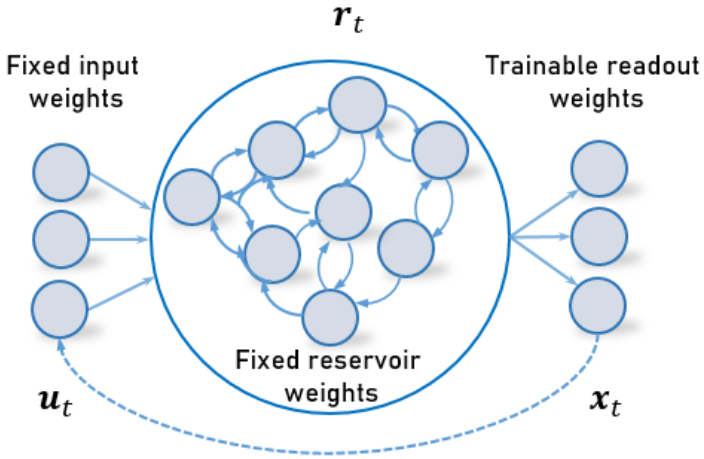

Figure 52: The basic principle of Reservoir Computing.

In RC, only the weights of the readout layer are trained, while the reservoir weights ($\mathbf{W}$ and $\mathbf{W}_{in}$) remain fixed. This approach significantly simplifies the training process. Since the weights involved in the dynamics do not receive any gradient updates, it avoids the challenges associated with BPTT discussed in Sect. 3.2.1. To spell this out, assume again we have observed a time series given by $(\boldsymbol{x}_t)_{t=1\ldots T}$, $\boldsymbol{x}_t \in \mathbb{R}^N$. This time series is iteratively fed into the reservoir as $\mathbf{u}_t$, where the reservoir state $\boldsymbol{r}_t$ is updated according to Eq. 101. The readout layer $\mathbf{W}_{out}$ connects reservoir states to observations $\hat{\boldsymbol{x}}_t = \mathbf{W}_{out}\boldsymbol{r}_t$. Since the reservoir states are already pre-computed, the readout weights can simply be determined by linear regression

$$\min_{\mathbf{W}_{out}} \sum_{t=1}^{T} \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|^2. \tag{102}$$

The RC can also be used to predict multiple steps by using its own predictions as input for the next time step, $\mathbf{u}_{t+1} = \hat{\boldsymbol{x}}_t$. Hence, RCs rely on the dynamic richness of the reservoir to capture the dynamics inherent in the observations. There is a large literature on initializing the reservoirs effectively to ensure this dynamical richness within the initialization. One critical aspect of this initialization is the topology of the reservoir [63, 79, 101, 145, 180, 435], where hub-like

networks or specific graph structures are often considered, similar in spirit to the ideas investigated by us in Hemmer et al. [149]. A second important component is the spectral radius of the reservoir weight matrix, which significantly influences the echo state property and the memory capacity of the reservoir. The spectral radius corresponds to the largest absolute eigenvalue of $\mathbf{W}$. A smaller spectral radius typically leads to a more stable, but less expressive network, as it constrains the dynamic range of the reservoir states, while a larger spectral radius can enhance the network's dynamics but can make optimization more unstable.

NEURAL ODES Neural Ordinary Differential Equations (Neural ODEs) [70] have emerged as a powerful framework for modeling continuous-time dynamics with NNs. The governing equation of a Neural ODE can be expressed as:

$$\frac{d\mathbf{z}(t)}{dt} = f_\theta(\mathbf{z}(t), t) \tag{103}$$

where $\mathbf{z}(t)$ denotes the state of the system at time t, and f_θ is a NN parameterized by θ that models the derivative of the state with respect to time. The choice of NN architecture is very flexible, including deep NNs and different nonlinear activation functions (for the results in [139], we for instance used a 4-layer Neural ODE). Neural ODEs fall into the class of models based on deep implicit layers [177]. Since the derivative $\frac{d\mathbf{z}(t)}{dt}$ is implicitly defined through an NN, this enables the deployment of different ODE solvers to integrate the system's dynamics. Therefore, the Neural ODE should be agnostic to the specific choice of numerical method for solving the differential equation, accommodating anything from simple Euler methods to more complex, adaptive-step methods. Using simple solvers can however lead to challenges in stiff ODE systems which exhibit rapid changes in parts of the state space [199] while using more sophisticated adaptive solvers can lead to higher computational costs. For the results in [152], we tested several fixed-step numerical solvers (`rk4`, `euler`, `midpoint`), which had little influence on the results, while an adaptive-step solver (`adaptive_heun`) led to prohibitively long training times.

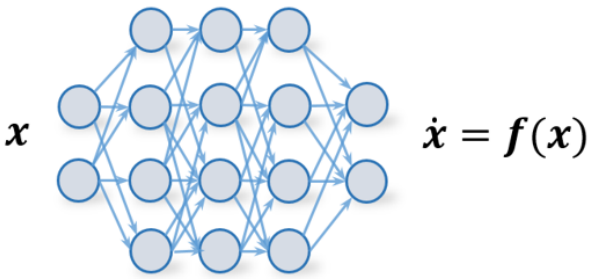

Figure 53: The basic principle of Neural ODEs.

As for most other DSR approaches, the parameters θ of the Neural ODE are learned by minimizing the MSE between observed data and model predictions. Gradients are computed based on the adjoint sensitivity method [70]. This method computes the solution of an auxiliary ODE backward in time to com-

pute gradients with respect to the parameters θ. Here, the adjoint state, $\mathbf{a}(t)$, is defined as the gradient of the loss L with respect to the state $\mathbf{z}(t)$:

$$\mathbf{a}(t) = \frac{\partial L}{\partial \mathbf{z}(t)}. \tag{104}$$

The evolution of the adjoint state backward in time is governed by

$$-\frac{d\mathbf{a}(t)}{dt} = \mathbf{a}(t)^\top \frac{\partial f_\theta(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)}. \tag{105}$$

The gradient of the loss L with respect to the parameters θ, which is ultimately of interest when training the model with gradient descent, is thus given by:

$$\frac{\partial L}{\partial \theta} = \int_{t_1}^{t_0} \mathbf{a}(t)^\top \frac{\partial f_\theta(\mathbf{z}(t), t)}{\partial \theta} dt, \tag{106}$$

where t_0 and t_1 represent the initial and final times of observation, respectively [70]. This formulation allows the computation of gradients without explicit knowledge of the solver's internal mechanics, thereby not restricting the choice of the numerical method for the forward pass, and does not require storing intermediate states during forward integration. It also means that Neural ODEs can naturally be applied to irregularly sampled time series [336]. For the comparisons presented in Chapter 4 we used an implementation based on the `torchdiffeq` package. Here we chose the number of layers and hidden sizes to make them comparable to the other models, while grid searching over different activation functions $\{elu, silu, tanh\}$, sequence length for training $\{5, 10, 25, 50\}$ used per batch, and learning rates $\{1e-3, 1e-2\}$.

LSTM-MSM Vlachas et al. [408] introduce a hybrid model that combines Long Short-Term Memory (LSTM) networks with Mean-Field Stochastic Models (MSM), based on Ornstein-Uhlenbeck processes. The LSTM component of the hybrid model is given by classical the governing equations of an LSTM [155], which is a combination of a hidden state, a cell state, a forget gate, and an input gate:

1. **Forget gate:** $f_t = \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f)$,
2. **Input gate:** $i_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i)$,
3. **Cell state update:** $\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_C)$,
4. **Final cell state:** $\boldsymbol{C}_t = f_t \cdot \boldsymbol{C}_{t-1} + i_t \cdot \tilde{\boldsymbol{C}}_t$,
5. **Output gate:** $o_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o)$,
6. **Hidden state:** $\boldsymbol{h}_t = o_t \cdot \tanh(\boldsymbol{C}_t)$.

Here, $\boldsymbol{W}$ and $\boldsymbol{b}$ denote the weight matrices and bias vectors, respectively, for each gate, σ is a sigmoid function, and tanh the hyperbolic tangent function. $\boldsymbol{h}_t$ and $\boldsymbol{C}_t$ are the hidden state and cell state at time t, and $\boldsymbol{x}_t$ an external input.

The MSM component comes into play when the forecasted trajectory diverges significantly from the training data. The model then transitions to a mean-field stochastic model, defined by an Ornstein-Uhlenbeck process:

$$dZ_i = c_i Z_i dt + \xi_i dW_i \tag{107}$$

where dW_i represents the increment of the Wiener process. The coefficients c_i and ξ_i are determined from the training data by estimating the data variance and decorrelation time directly from the data. The LSTM component of the model is trained using truncated BPTT. Truncated BPTT limits the number of time steps over which gradients are propagated backward, thus avoiding the problems discussed in Sect. 3.2.1 by restricting the sequence length T_{seq} directly.

A.2.2 *Multimodal Comparisons*

CLASSICAL RNN TRAINING 'Classical' RNN training is still probably the most common technique for training RNN models in the literature. Here the model is not freely iterated forward from some initial state, but, much as with RCs, observations are provided as external inputs to the model at every time step. To freely generate long time series after training, the model can be provided its predictions at time t as input for the predictions at time $t+1$. Given some observed (potentially multimodal) time series $\boldsymbol{Y}$ of length T, observations $\boldsymbol{y}_t$ are included in the latent equation of a DSR model, e.g. a PLRNN (compare Eq. 12) via an input-to-hidden factor loading matrix $\boldsymbol{C}$:

$$\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{z}_{t-1} + \boldsymbol{W}\phi(\boldsymbol{z}_{t-1}) + \boldsymbol{C}\boldsymbol{y}_t + \boldsymbol{h} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}) \tag{108}$$

After obtaining a latent trajectory $\boldsymbol{Z}$, the model can, as during training with MTF, be optimized using modality-specific decoder models. Training is performed on short subsequences of length T_{seq}, where T_{seq} is a hyperparameter, as in truncated BPTT. We found similar values for T_{seq} as for the sparse TF interval τ to work well in practice.

MULTIPLE SHOOTING An alternative approach to training RNN models that control gradient divergence is multiple shooting (MS) [46], a method already introduced in the context of DSR [411]. To train an RNN with MS, an observed time series, consisting of unimodal or multimodal observations $\boldsymbol{Y}$, is divided into N_{seq} subsequences $\boldsymbol{Y}^s, s = 1 \dots N_{seq}$, each of length L. Then every subsequence is treated as an individual initial value problem, where the initial conditions (shooting nodes), are learned as free model parameters. Coherence between different subsequences is achieved by enforcing the continuity between model-predicted states and shooting nodes with an explicit loss term. During training, trajectories for each subsequence are autonomously generated from the initial states $\boldsymbol{\mu}_0^s$ over L time steps, using a DSR model F_θ (Eq. 22). Then, the last predicted time step $F_\theta(\boldsymbol{z}_L^s)$ is compared with the next shooting node $\boldsymbol{\mu}_0^{s+1}$, and their consistency is enforced via a loss term scaled by a regularization constant λ_{MS}:

$$\mathcal{L}_{MS} = \lambda_{MS} \sum_{s=1}^{N_{seq}-1} \|F_\theta(\boldsymbol{z}_L^s) - \boldsymbol{\mu}_0^{s+1}\|_2^2 \tag{109}$$

Here, $F_\theta(\boldsymbol{z}_L^s) = F_\theta(F_\theta(\dots F_\theta(\boldsymbol{\mu}_0^s))) = F_\theta^L(\boldsymbol{\mu}_0^s)$. From the generated latent trajectories, the likelihoods of the observations $\boldsymbol{Y}^s$ can be evaluated, using modality-specific decoder models (Sect. 3.2.6). As MS does not require any inversion of the observation model, it naturally accommodates multiple observations. However, since it does not co-train an encoder model, it can not be leveraged to

make short-term predictions, since the model can not be initialized from new observations (see Sect. 4.1). Further note that the sequence length L plays a similar role as the TF interval τ for MTF, controlling the times at which states and gradients are reset during training. Optimal settings for τ and L agreed during experiments.

GAUSSIANIZATION OF NON-NORMAL MODALITIES Since the TF-based approaches discussed in 3.2.3 and 3.2.4 require the inversion of the observation model to obtain control states, which is, as in the case of id-TF, only straightforward if observations are normally distributed, another approach for training on multimodal data involves simply preprocessing all modalities to align them with Gaussian assumptions. For the comparisons in 4.3.1, we applied a Box-Cox transformation [48], z-scoring, and Gaussian kernel smoothing to transform ordinal and count observations into variables that approximately follow a Gaussian distribution. To determine the most effective Gaussian kernel width, we conducted a grid search across a range of kernel sizes [53].

A.3 DATASETS

A.3.1 *Benchmark Systems*

LORENZ-63 SYSTEM The 3d chaotic Lorenz attractor, suggested in Lorenz [252] is probably the most famous chaotic dynamical system in the literature and the most widely used benchmark system in DSR. Its governing equations are given by:

$$\begin{aligned} dx &= (\sigma(y-x))dt + d\epsilon_1(t), \\ dy &= (x(\rho-z)-y)dt + d\epsilon_2(t), \\ dz &= (xy-\beta z)dt + d\epsilon_3(t). \end{aligned} \tag{110}$$

Standard parameter settings putting the system into a chaotic regime are $\sigma = 10, \rho = 28$, and $\beta = 8/3$. This formulation as a stochastic differential equation (SDE) includes process noise, which we injected into the system when drawing trajectories by adding a random Gaussian noise term $d\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.01^2 dt \times \mathbf{I})$.

RÖSSLER SYSTEM The chaotic Rössler system from Rössler [343] is defined by:

$$\begin{aligned} dx &= (-y-z)dt + d\epsilon_1(t), \\ dy &= (x+ay)dt + d\epsilon_2(t), \\ dz &= (b+z(x-c))dt + d\epsilon_3(t). \end{aligned} \tag{111}$$

For the benchmarks reported in 4.2, we used $a = 0.2, b = 0.2$, and $c = 5.7$. Process noise was included as before by drawing $d\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.01^2 dt \times \boldsymbol{I})$.

BURSTING NEURON MODEL The 3-dimensional bursting neuron model from Durstewitz [94] is defined as a combination of slow and fast variables:

$$\begin{aligned} -C_m\dot{V} = {} & g_L\,(V-E_L) + g_{Na}\,m_\infty(V)\,(V-E_{Na}) \\ & + g_K n\,(V-E_K) + g_M h\,(V-E_K) \\ & + g_{NMDA}\left[1+.33e^{-.0625V}\right]^{-1}(V-E_{NMDA}) \end{aligned} \tag{112}$$

$$\dot{n} = \frac{n_\infty(V)-n}{\tau_n},\quad \dot{h} = \frac{h_\infty(V)-h}{\tau_h}, \tag{113}$$

where

$$\{m_\infty, n_\infty, h_\infty\} = \left[1+e^{(\{V_{hNa},V_{hK},V_{hM}\}-V)/\{k_{Na},k_K,k_M\}}\right]^{-1}. \tag{114}$$

For the results in Sect. 4.2, we used the same parameter settings from Schmidt et al. [355] to shift the system into a complex but not chaotic limit cycle, where:

$$\begin{aligned} &C_m = 6\mu F, g_L = 8mS, E_L = -80mV, g_{Na} = 20mS, \\ &E_{Na} = 60mV, V_{hNa} = -20mV, k_{Na} = 15, g_K = 10mS, \\ &E_K = -90mV, V_{hK} = -25mV, k_K = 5, \tau_n = 1\text{ ms}, g_M = 25mS \\ &V_{hM} = -15mV, k_M = 5, \tau_h = 200\text{ ms}, g_{NMDA} = 10.2mS \end{aligned}$$

WILSON COWAN MODEL The Wilson-Cowan model [422] is a popular model of neural population dynamics. It describes interactions between a collection of excitatory (E) cells and inhibitory (I) cells, described by:

$$\tau_i \frac{dr_i}{dt} = -r_i + \phi\,(w_{ei}\cdot r_e - w_{ii}\cdot r_i - z_i) \tag{115}$$

$$\tau_e \frac{dr_e}{dt} = -r_e + \phi\,(w_{ee}\cdot r_e - w_{ei}\cdot r_i - z_e)\,, \tag{116}$$

Here, $w_{ei}, w_{ee}, w_{ie}, w_{ii}$ are coupling strengths, z_i and z_e denote constant input currents, and τ_i and τ_e are time constants. For the simulations in [54], I chose parameter settings that placed the model into a multistable regime, with two stable fixed points and one unstable fixed point: $w_{ei} = 9., w_{ee} = 9., w_{ie} = 5., w_{ii} = 5., z_e = 3, z_i = 4$. The vector field and fixed points for this configuration are shown in Fig. 20.

LORENZ-96 SYSTEM The Lorenz-96 [253] is a spatially extended weather model, featuring interaction terms between neighboring units, and extending characteristic spatiotemporal dynamical features:

$$dx_i = ((x_{i+1}-x_{i-2})x_{i-1} - x_i + F)dt + d\epsilon,\ i = 1\ldots N, \tag{117}$$

Here F is a constant forcing term, where $F = 8$ was used in the experiments, leading to chaotic behavior. As for the other chaotic benchmarks, process noise was injected with $d\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.01^2 dt \times \mathbf{I})$.

LEWIS-GLASS CHAOTIC NETWORK MODEL The Lewis-Glass model is a 6d network model introduced in Lewis and Glass [229]. The vector field is given by

$$\frac{d\boldsymbol{x}}{dt} = \frac{-\boldsymbol{x}}{\tau} + G(\epsilon\,\boldsymbol{K}\boldsymbol{x}) - \beta. \tag{118}$$

with activation function $G(x) = \frac{1+\tanh(-\alpha x)}{2}$, and $\boldsymbol{K}$ a specific connectivity matrix, given by

$$\boldsymbol{K} = \begin{bmatrix} 0 & -1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & 0 & 0 & -1 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 \end{bmatrix}$$

and $\alpha = -1, \beta = 0.5, \epsilon = 10, \tau = 2.5$. The implementation of the model was based on the `dysts.flows` package from Gilpin [125].

NEURAL POPULATION MODEL As a benchmark for more complex large-scale chaotic dynamics in [54], we used the chaotic network model from Landau and Sompolinsky [224], which combines structured connectivity (a rank-1 component) with a randomly initialized network matrix. Specifically, the network dynamics are given by

$$\frac{d\mathbf{h}}{dt} = -\mathbf{h} + J\phi(\mathbf{h}) + \frac{J_1}{\sqrt{N}}\xi v^T \phi(\mathbf{h}), \tag{119}$$

where $\phi(\mathbf{h}) = \tanh(\mathbf{h}(t))$. For the reconstructions in Sect. 4.2.2, we sampled trajectories from a 50-d network model with $J_1 = 0.09$.

FORCED DUFFING SYSTEM The forced Duffing equation represents a nonlinear oscillator with a double-well potential, introduced in Hamel [142]. The forced version of the system is usually written as:

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma\cos(\omega t), \tag{120}$$

with linear stiffness coefficient α, non-linear stiffness coefficient β, damping coefficient δ, forcing amplitude γ, and forcing frequency ω. For the chaotic regime often explored in the literature and used for the results in Sect. 4.5, parameters were set to $\alpha = -1, \beta = 1, \delta = 0.3, \gamma = 0.37$, and $\omega = 1.2$.

The unforced Duffing system is usually formulated as a set of two coupled ODEs:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= ay - x\left(b + cx^2\right) \end{aligned} \tag{121}$$

where $[a, b, c] = [-\frac{1}{2}, -1, \frac{1}{10}]$ places the system into a multistable regime with two coexisting EPs (see Figs. 6 and 51).

A.3.2 *Experimental Datasets*

EEG DATASET The Electroencephalogram (EEG) dataset used in ([54, 152], Sect. 4.2.2) was taken from a study in [352], comprising 64-channel EEG data from human subjects engaged in various motor and imagery tasks. For the evaluations, we focused on the 'eyes open' baseline time series from subject

0, since this featured the least amount of movement artifacts, and consisted of a total of 9760 time steps. Before training the signal was standardized and smoothed using a Hann function.

ECG DATASET The Electrocardiogram (ECG) data was taken from the publically available PPG-DaLiA dataset [328], captured using a chest-worn device with a sampling rate of 700 Hz. For the experiments in ([54, 152], Sect. 4.2.2), we used the first one-dimensional time series for subject 2 in a sitting position with a total of 419973 time steps. The time series was again standardized and smoothed with a Gaussian kernel ($\sigma = 5$ time bins). We further included a temporal delay embedding. In [54], we used a delay dimension $m = 7$ and lag $\tau_{lag} = 61$, determined using the `DynamicalSystems.jl` Julia package [82]. In [152], we chose delay-embeddings based on the PECUZAL algorithm [215], likewise implemented implemented in `DynamicalSystems.jl`. The algorithm employs non-uniform delays with different time lags to find an optimal delay embedding, using a Theiler window based on the first minimum of the mutual information. This led to an embedding dimension of $m = 5$.

HUMAN FUNCTIONAL MAGNETIC RESONANCE IMAGING (FMRI) DATASET This dataset was taken from Kramer et al. [214], where it was studied in a similar context for multimodal data integration in DSR, and originally taken from [209]. The dataset consisted of 26 participants, out of which I selected 20 participants, which did not feature strong movement artifacts which often lead to unstable training due to strong discontinuities in the observed time series. Participants were shown a series of images (rectangles and triangles) while undergoing fMRI recording. The sampling rate was 1/3 Hz. The neural activity was assessed via the blood oxygenation level-dependent (BOLD) signal, projected on the first principal component, in 10 different brain regions per hemisphere identified to be relevant for the tasks. The subjects then had to carry out several different cognitive tasks concerning the displayed shapes: a continuous delayed response 1-back task (CDRT), a continuous matching 1-back task (CMT), and a 0-back control choice reaction task (CRT). All tasks were repeated five times, with a resting condition in between tasks and an instruction phase before each task. We excluded the last repetition as a test set for assessing predictive performance.

HIPPOCAMPAL MULTIPLE SINGLE-UNIT (MSU) AND SPATIAL POSITION DATA As the second empirical multimodal dataset, we used electrophysiological recordings from the rodent hippocampus, combined with spatial location data [133] publicly accessible at `https://crcns.org/data-sets/hc/hc-11/about-hc-11`. Our analysis focused on the recordings of the rat 'Achilles'. The extracted spike times were preprocessed using the script provided in [445], to obtain counts per time interval with a bin width of 200 ms, and selected the 60 most active neurons. Since for our analysis, we were primarily interested in the combination of spike counts and position data, we focused solely on the MAZE task, in which the rat was moving along a platform, with the rats receiving water rewards at both ends of the track, represented in our model as scalar external inputs s_t (cf. equation 12), set to 1 for five time bins when the rat moved away from the reward location.

B

APPENDIX RESULTS

B.1 FURTHER RESULTS

Table 7: Comparisons of SOTA DSR methods on the Lorenz-63 and Lorenz-96 system. Note that our training methods perform approximately on par with SINDy, which features a strong inductive bias in favor of reconstructing the two benchmark systems since these are low-order polynomials which the provided library included in these experiments (see Sect. A.2.1). Taken from [152].

Dataset	Method	$D_{stsp} \downarrow$	$D_H \downarrow$	PE(20) $\downarrow$	dim	$\lvert\boldsymbol{\theta}\rvert$
Lorenz-63 (3d)	shPLRNN + GTF	$\mathbf{0.26 \pm 0.03}$	$\mathbf{0.090 \pm 0.007}$	$\mathbf{(6.0 \pm 0.5) \cdot 10^{-4}}$	3	365
	dendPLRNN + id-TF	0.9 ± 0.2	0.15 ± 0.03	$(2.2 \pm 0.7) \cdot 10^{-3}$	10	361
	RC	0.52 ± 0.12	0.19 ± 0.04	$(5 \pm 2) \cdot 10^{-3}$	201	603
	LSTM-TBPTT	0.46 ± 0.22	0.11 ± 0.03	$(1.1 \pm 0.3) \cdot 10^{-3}$	30	1188
	SINDy	$\mathbf{0.24 \pm 0.00}$	$\mathbf{0.091 \pm 0.000}$	$\mathbf{(6.1 \pm 0.0) \cdot 10^{-4}}$	3	30
	N-ODE	0.63 ± 0.2	0.15 ± 0.05	$(2.3 \pm 0.3) \cdot 10^{-3}$	3	353
	LEM	0.39 ± 0.24	0.12 ± 0.05	$(6.0 \pm 0.9) \cdot 10^{-3}$	14	360
Lorenz-96 (20d)	shPLRNN + GTF	1.68 ± 0.06	$\mathbf{0.072 \pm 0.001}$	$(1.21 \pm 0.02) \cdot 10^{-1}$	20	4540
	dendPLRNN + id-TF	$\mathbf{1.65 \pm 0.05}$	0.083 ± 0.005	$\mathbf{(1.1 \pm 0.1) \cdot 10^{-1}}$	60	5740
	RC	2.40 ± 0.15	0.14 ± 0.02	$(4.9 \pm 0.4) \cdot 10^{-1}$	600	12000
	LSTM-TBPTT	5 ± 1	0.31 ± 0.04	$(1.14 \pm 0.04) \cdot 10^{0}$	80	10580
	SINDy	$\mathbf{1.59 \pm 0.00}$	$\mathbf{0.06 \pm 0.00}$	$\mathbf{(4.6 \pm 0.0) \cdot 10^{-3}}$	20	4620
	N-ODE	1.77 ± 0.07	0.076 ± 0.01	$(2.5 \pm 0.02) \cdot 10^{-1}$	20	4530
	LEM	7.2 ± 2.3	0.54 ± 0.13	$(1.3 \pm 0.06) \cdot 10^{0}$	46	4620

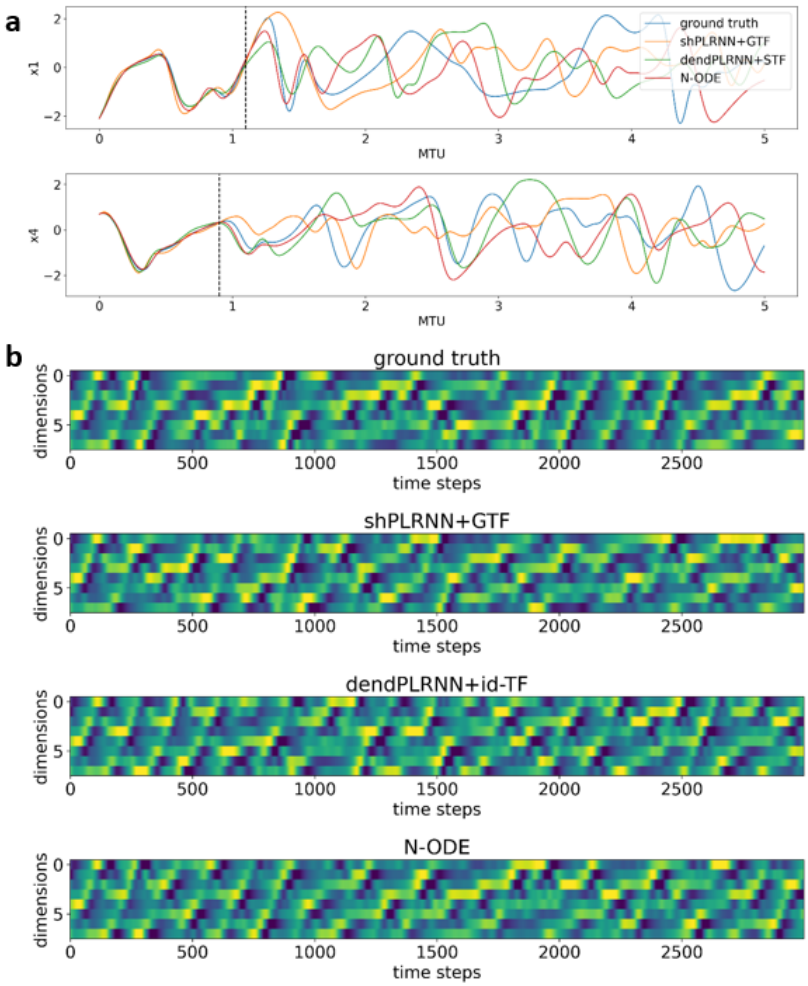

Figure 54: Short-term predictions (**a**) and long-term spatiotemporal behavior (**b**) for our models and N-ODEs for the multiscale Lorenz-96 system, introduced in [68] to assess forecasting performance of different DSR approaches. Taken from [152].

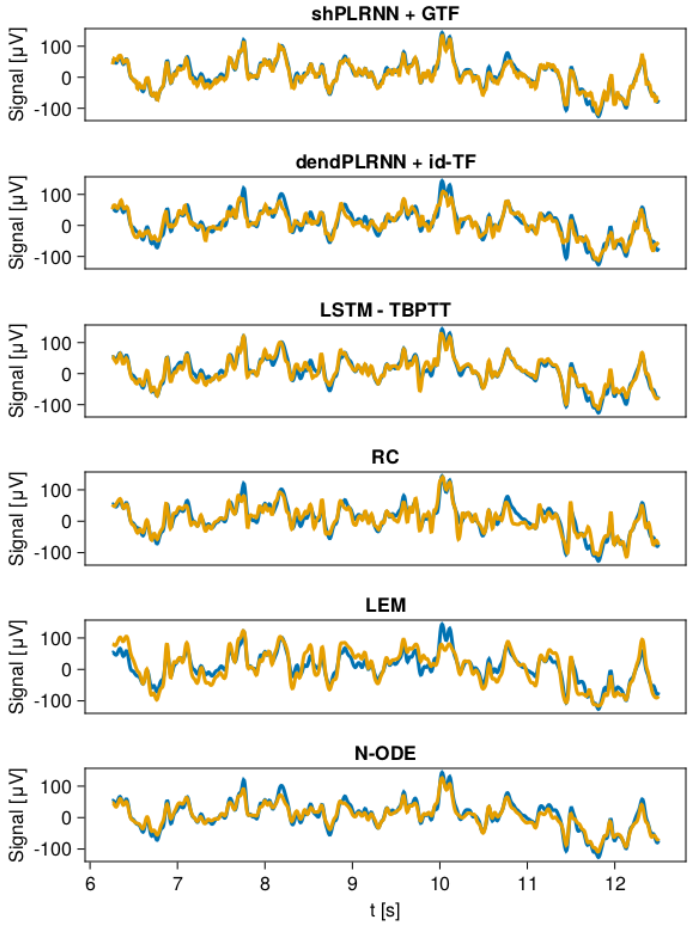

Figure 55: 5-step-ahead predictions (yellow) of DSR algorithms on EEG time series (blue). While all methods provide reasonable short-term forecasts, reflecting the optimization criterion, most fail to produce non-trivial limiting dynamics (Fig. 23). Taken from [152].

Table 8: Comparison of dendPLRNN trained by MTF with the MVAE [214], and an MS approach, on 8 ordinal observations with seven ordered categories, produced by the chaotic Lorenz system, Rössler system, and Lewis-Glass model, and on a symbolic representation of the chaotic Lorenz system. Values are mean ± SEM, averaged over 15 trained models. Taken from [53].

Dataset	Method	D_{stsp} ↓	λ_{max}	OPE ↓	SCC ↓	OACF ↓
Lorenz-ordinal	MTF	**8.8 ± 0.59**	**0.92 ± 0.39**	**0.24 ± 0.015**	**0.085 ± 0.02**	**0.016 ± 0.04**
	SVAE	14.7 ± 0.7	0.44 ± 0.71	0.8 ± 0.03	0.17 ± 0.02	0.23 ± 0.02
	MS	13.8 ± 1.1	0.47 ± 0.67	X	0.24 ± 0.06	0.15 ± 0.03
Rössler-ordinal	MTF	**7.9 ± 0.8**	**0.03 ± 0.07**	**0.093 ± 0.007**	**0.051 ± 0.009**	**0.051 ± 0.009**
	SVAE	11.5 ± 1.3	-0.27 ± 0.58	0.39 ± 0.02	0.23 ± 0.05	0.18 ± 0.04
	MS	14.1 ± 1.0	-0.05 ± 0.12	X	0.12 ± 0.04	0.14 ± 0.03
Lewis-Glass-ordinal	MTF	**0.89 ± 0.04**	**−0.11 ± 0.41**	**0.15 ± 0.02**	**0.28 ± 0.05**	**0.15 ± 0.03**
	SVAE	1.40 ± 0.22	−1.8 ± 2.1	0.29 ± 0.01	0.49 ± 0.04	0.24 ± 0.02
	MS	1.0 ± 0.14	−0.14 ± 0.31	X	0.51 ± 0.04	0.45 ± 0.03
Lorenz-symbolic	MTF	**4.4 ± 2.69**	**0.67 ± 0.37**			
	SVAE	12.02 ± 1.98	1.87 ± 0.88			
	MS	**4.46 ± 1.82**	5.67 ± 1.25			

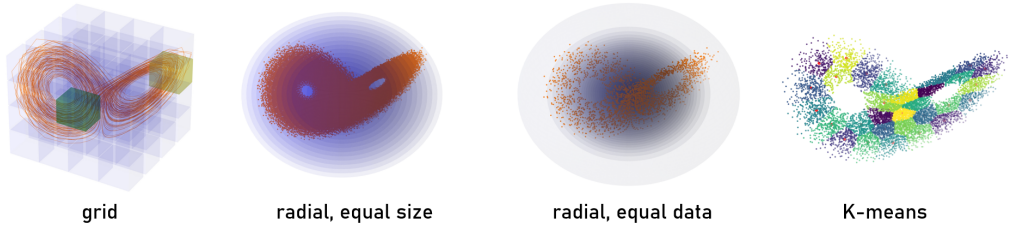

Figure 56: Different approaches for symbolizing the same underlying dynamical system. In preliminary experiments, MTF managed to approximately reconstruct the underlying DS from all four symbolic representations.

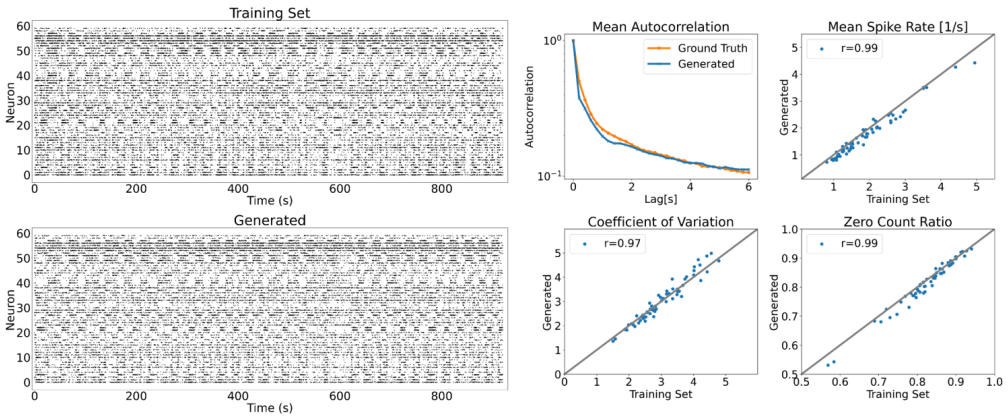

Figure 57: Example reconstructions of spike trains and spike statistics on the training set (see Methods A.3.2). Test set reconstructions and further statistics are in Figure 32. Taken from [53].

LIST OF FIGURES

Figure 1 Illustration of chaos at the example of two simulated solutions of the chaotic Lorenz-63 system (Eq. 110) using the same numerical solver (Runge-Kutta method of order 5(4) [87] from `scipy.integrate`), with the sole exception being a small difference in absolute error tolerances (10^{-8} vs. 10^{-11}) of the numerical integrator. While trajectories diverge rapidly after a certain time horizon, the long-term limit sets still closely resemble each other. 17

Figure 2 Flow fields and example trajectories (blue) for different 2D and 3D attractors. 17

Figure 3 Overview over dynamical systems reconstruction. 19

Figure 4 Illustration of temporal delay embeddings. 20

Figure 5 Illustration of measurement functions in DSR. 21

Figure 6 Generalization and multistability in DSR. 28

Figure 7 The dendPLRNN extends each neuron of the PLRNN into a set of nonlinear branches, significantly increasing its expressivity and enabling reconstructions in lower dimensions. Figure created with the artistic support of Darshana Kalita. Taken from [54]. 37

Figure 8 Basis expansion reduces latent space dimensionality. **a**: Agreement in attractor geometries (Sect. 4.1) (top) and proportion of successful reconstructions (bottom) for the Lorenz-63 system as a function of the number of bases (B) and latent states (M). $B = 0$ in the top graph denotes the standard PLRNN (no basis expansion). **b**: Runs with $D_{stsp} < 4$ were defined as successful (similar results are obtained with other choices for the D_{stsp} threshold). Taken from [54]. 39

Figure 9 Illustration of the SVAE setup. 44

Figure 10 Principle of Generalized Teacher Forcing. Taken from [152]. 48

Figure 11 Example training sequences ($T_{seq} = 300$) at different stages of training using STF ($\tau = 25$, with forcing times highlighted in red), and GTF ($\alpha = 0.15$) for training a shPLRNN on the Lorenz-63 system. Forced states with STF do not align perfectly with the data since forcing occurs prior to the RNN step. Note that training on this sequence length without any TF quickly leads to divergences. Hence the unforced prediction (light green) is drawn from the model trained with GTF, and intended to serve as a reference for the freely evolving model predictions at this epoch. 51

Figure 12 Principle of Multimodal Teacher Forcing. Taken from [53]. 52

Figure 13 Illustration of the impact of dramatic changes in long-term dynamics on the SVAE and MTF loss. **a**: A dendPLRNN successfully trained on multimodal observations from the Lorenz-63 system is altered by setting a parameter of the linear self-connectivity $A_{22} > 1$, which results in globally diverging dynamics, while still looking locally consistent with the Lorenz-63. **b**: The global divergence is reflected in the training-time trajectories $\boldsymbol{Z}$ generated using MTF with interval $\tau = 10$ (right), within which the DSR model evolves freely. This divergence leads to large increases in the MTF training loss (see MTF loss curve for $A_{22} > 1$), and hence is strongly penalized. This effect is essentially not present for the SVAE, where the global divergence induces no considerable effect on the training loss and training-time trajectories. The mismatch in global (long-term) dynamics hence remains unrecognized by the SVAE. As shown at the bottom, at the minimum of the SVAE loss ($A_{22} \approx 0.966$) the dynamics converge to an equilibrium point (left), while MTF at its minimum ($A_{22} \approx 0.637$) produces trajectories which agree in their temporal structure with those of the original Lorenz-63 (right). 54

Figure 14 **a**: MTF loss landscapes, computed using the total loss (Eq. 60) by varying two parameters of a trained dendPLRNN (A_{22}, W_{34}) and computing the loss for a sequence of $T = 300$ time steps. Illustrated are four values of TF interval τ. Lower values of τ increasingly smoothen the loss landscape. $\tau = 10$ corresponds to an optimal choice for the TF interval, where the loss landscape appears both smoothed out and convex, while for low $\tau = 1$, the loss landscape flattens, making training more difficult. **b**: Comparison of MTF and SVAE loss landscapes. Since the SVAE loss (Eq. 33) only includes one-step ahead predictions from the DSR model, it essentially over-smoothens the loss landscape, similar to the observations made when choosing a very small τ in MTF, not allowing the model to evolve freely during training. Note that the parameter range that can be meaningfully explored for the MTF is smaller than for the SVAE since larger variations in the parameters (e.g. a value of A_{22} over 1) induce divergences in the sequences drawn from the DSR model for the computation of the DSR loss (see also Fig. 13). Based on [53]. 55

Figure 15 Comparison of state space agreement D_{stsp} in two scenarios (see Sect. 4.1 for details): (**a**) when omitting different terms from the total loss as specified in Eq. 60, and (**b**) while altering the scaling of the consistency loss $\mathcal{L}_{con}$. These comparisons are made for a dendPLRNN trained using MTF on multimodal Gaussian, ordinal, and count data from the chaotic Lorenz system (Fig. 24). Taken from [53]. 56

Figure 16 Illustration of the hierarchical inference framework. 62

Figure 17 Overview over metrics for continuous data for models trained on the chaotic Lorenz-63 system. **a**: While short-term predictions deteriorate after a certain time horizon for chaotic systems due to exponential trajectory divergence (here after around 120 time steps), the long-term temporal patterns can still agree (right). **b**: State-space agreement approximated via the binning approximation, projected along the z-direction. **c**: Correlation between D_{stsp} approximated via the binning method ($m = 30$) and the logarithm of D_{stsp} approximated as a GMM for generated data from different trained models. **d**: Example power spectra for different smoothing factors σ. (**c**) and (**d**) taken from [54]. 66

Figure 18 Illustration of the state space measure in the absence of continuous observations. 70

Figure 19 **a**: Ground truth and rotated attractors of the Rössler system with associated D_{stsp}-values. **b**: Correlation between D_{stsp} for models trained on trajectories from the Rössler system, computed directly in observation space given a co-trained linear (Gaussian) observation model (D_{bin}), and after approximation applying PCA and the combined rotation operation directly in the 20-dimensional state space (D_{PCA}), based on a total of 30 trained models. Based on [53]. 71

Figure 20 Reconstructed and ground truth vector field for the 2D Wilson-Cowan system, including locations of the analytically obtained fixed points the trained dendPLRNN, and ground truth fixed point locations. Taken from [54]. 75

Figure 21 Example reconstructions of two low-dimensional benchmark systems, produced from a dendPLRNN trained with VI ($B = 20$, $M = 15$) on the chaotic Lorenz system (**a**, Eq. 110) and with STF ($B = 47$, $M = 26$, $\tau = 5$) on the bursting neuron model (**b**, Eq. 112), with time series (left) and state space reconstructions for both true and generated time series (right). Since the bursting neuron model is non-chaotic, model predictions agree closely with the ground truth data for up to 1000 time steps, while due to the chaotic nature of the Lorenz system, predictions diverge while agreeing in terms of overall temporal and geometric structure. Taken from [54]. 76

Figure 22 Example reconstruction of DSR from high-dimensional benchmark systems, using a dendPLRNN trained with STF ($B = 50$, $M = 30$, $\tau = 10$). **a**: Time series (top), spatiotemporal evolution (middle), and power spectra (bottom) for the true 10d Lorenz-96 system (Eq. 117) and for time series sampled from the dendPLRNN. **b**: Same for a 50d neural population model (Eq. 119) ($B = 5$, $M = 12$, $\lambda = 1.0$, $M_{reg}/M = 0.2$). Taken from [54]. 78

Figure 23 Example time traces of ECG (**a**) and EEG (**b**) reconstructions for all methods compared in Table 4. For each model, we picked the best run out of 20 runs, according to the state space agreement D_{stsp}. Taken from [152]. 80

Figure 24 **a**: Sample trajectories and time series produced by a dendPLRNN with parameters ($M = 20$, $B = 10$, $K = 15$, $\tau = 10$), trained using MTF on multimodal data (Gaussian, ordinal, and count)—sampled from a Lorenz-63 system. **b**: Example power spectra from Gaussian data alongside Spearman autocorrelation functions for ordinal and count data. Taken from [53]. 82

Figure 25 DSR from multimodal observations (continuous and ordinal, sampled from a Lorenz-63 system) using MTF, where continuous observations were distorted with medium (10% of data variance) (**a**) and high (50% of data variance) noise levels (**b**-**c**). **a**: Freely generated example trajectories from a dendPLRNN ($M = 20, B = 10, K = 20, \tau = 10$). **b**: Same as (**a**), but for heavily distorted Gaussian observations. The maximum Lyapunov exponent (λ_{max}) of the dendPLRNN resembles that of the GT system, $\lambda_{max} \approx 0.903$). **c**: Normalized cumulative histograms of geometrical attractor disagreement (D_{stsp}, left) and Hellinger distance (D_H, right) between reconstructed and ground-truth system with and without ordinal observations indicate that DSR from highly distorted data in the unimodal case is impossible. Taken from [53]. 83

Figure 26 Cross-modal inference, using the mixture-of-experts encoder model. **a**: Reconstructions of the Lorenz-63 from Gaussian and ordinal observations, where 20% of time steps are missing at random time points individually drawn for each modality. This allows the model to develop useful cross-modal links and infer an approximate posterior estimate even when observations from the other modality are missing. **b**: Using only the Gaussian expert to encode ground truth Gaussian observations, the corresponding ordinal ratings can be almost perfectly decoded, including steps missing in the ordinal training data. 84

Figure 27 Encoded states $p(\tilde{\boldsymbol{Z}}|\boldsymbol{X})$ (left) after training a shPLRNN with MTF on one-dimensional observations of the Lorenz-63 system (x coordinate, bottom), resemble the unfolded attractor using a temporal delay embedding (right) with optimal settings determined using the minimum of the mutual information [187]. For this plot, states were again approximately overlayed with the delay-embedded states using Eq. 75. 85

Figure 28 DSR from discrete observations for the Rössler system ($\lambda_{\text{max}}^{\text{true}} \approx 0.072$) and Lorenz system $\lambda_{\text{max}}^{\text{true}} \approx 0.903$). Note that in all cases the topology and general geometry are preserved, and maximal Lyapunov exponents closely match those of the true systems. First row: Symbolic coding of Lorenz attractor (see Fig. 30 for true and predicted class label probabilities and statistics), TDE = temporal delay embedding. Second row: Reconstruction of Rössler attractor from 8 ordinal time series with 7 levels each. Third row: DSR of Lorenz attractor from 8 ordinal time series with 7 levels each. Fourth row: DSR of Rössler attractor from 60 ordinal time series with 2 levels each. Based on [53]. 86

Figure 29 Reconstruction of a 10d chaotic Lorenz-96 system solely from ordinal observations with up to 15 levels using a shPLRNN ([152]; $M = 10, L = 100, \tau = 10$). **a** top: Ground truth ordinal time series sampled from a randomly initialized ordinal observation model $p(\boldsymbol{o}_t|\boldsymbol{x}_t)$ from ground truth states $\boldsymbol{x}_t$ of the Lorenz-96 system, and reconstructed ordinal observations decoded from freely generated latent states using the trained decoder model $p(\boldsymbol{o}_t|\tilde{\boldsymbol{z}}_t)$. Bottom: Example ground truth and freely generated ordinal time series from 1 channel. **b**: Ground truth states $\boldsymbol{x}_t$ (top), states encoded using the trained MTF encoder $p(\tilde{\boldsymbol{z}}_t|\boldsymbol{o}_t)$ (center), and *freely generated* latent activity from the trained DSR model $\boldsymbol{z}_t = F_\theta(\boldsymbol{z}_{t-1})$ (bottom). States $\tilde{\boldsymbol{z}}_t$ encoded from the ordinal data were aligned with ground truth states $\boldsymbol{x}_t$ (not seen during training) using a linear operator $\boldsymbol{B}$ (Eq. 75). This linear operator was also used to project the freely generated activity of the shPLRNN into the observation space of the Lorenz-96 system. **c** left: Example of ground truth (orange) and freely generated (blue) activity. **c** right: Aligned ground truth ($\boldsymbol{x}_t$) and encoded latent states ($\tilde{\boldsymbol{z}}_t$) as in **b** for one example unit. Note that the encoded states $p(\tilde{\boldsymbol{z}}_t|\boldsymbol{o}_t)$ and ground truth states $\boldsymbol{x}_t$ overlap almost perfectly, although the $\boldsymbol{x}_t$ have never been seen by the model during training. 87

Figure 30 **a**: True and predicted class label probabilities (given the maximum posterior probability for a category at each time step) from a freely generated trajectory of and dendPLRNN, trained with MTF on the symbolic representation of the chaotic Lorenz-63 dynamics. **b** and **c**: Kernel-density estimates of maximum Lyapunov exponents (**b**) and cumulative distributions of D_{stsp} (**c**), comparing training with MTF, Multiple Shooting (MS), and sequential multimodal VAE (MVAE) across 30 trained models each. Taken from [53]. 88

Figure 31 **a**: Multimodal integration on functional magnetic resonance imaging (fMRI)+behavioral data significantly improves DSR compared to just training on fMRI data alone (unimodal). Results are shown for 20 subjects (subjects represented by black lines, with the mean across subjects by a blue line), shown for both geometrical (D_{stsp}, left) and temporal (D_H, right) disagreement between true and reconstructed systems. p-values obtained by performing a paired t-test. **b**: Example of decoded (color-coding of time series) and true (background colors) task stages $\hat{l} \in$ {Rest, Instruction, CRT, CDRT, CMT} for an example subject. The trained model was freely iterated forward from the first time step of the test set not seen during training, and task stages were decoded from the simulated activity based on the maximum posterior probability, $\hat{l}_t = \arg\max p(l_{kt}|z_t)$, given the latent trajectory z_t. **c**: Example subspaces of freely generated latent activity for a DSR model trained jointly on continuous and categorical data by MTF for an example subject. Task labels at each latent state are predicted according to the maximum posterior probability given the latent state at each time step, as in **b**. The latent space is structured according to the task stages. **d**: Freely generated time series from 10 brain areas per subject from subjects #3 (left) and #7. (right). The trained DSR model, only iterated by providing an initial state, captures the overall temporal structure of the complex activity patterns even from very short time series. Based on [53]. 90

Figure 32 **a**: Example reconstructions of spike trains and spatial location of a rat moving along a vertical track on the unseen test set (second half of the trial), generated from a data-inferred initial condition. **b**: Correlation of mean spike rate, zero count ratio, coefficient of variation, and correlation between cross-correlation coefficients between all 60 reconstructed neurons between test set and model-generated data (blue), and between experimental training and test set data (orange). Diagonal gray lines are bisectrices. Bottom: Cross-correlation matrices among all 60 neurons for the test set (left) and model-generated data (right). **c**: Joint DSR from both spatial and neural data significantly improves reconstructions across all spike statistics, as assessed by computing the average MSE between spike statistics across all neurons. The MSE was normalized for each statistic for better visibility. **d**: Subspace of the DSR model's latent space, illustrating how the latent dynamics are structured according to the animal's spatial position. Based on [53]. 92

Figure 33 **a**: Reconstructions of the Lorenz-63 attractor (left) and Rössler attractor (right) for different values of ρ and c, respectively, using a hierarchical shPLRNN with a one-dimensional feature vector. **b**: By performing linear regression on the one-dimensional feature vectors $\boldsymbol{l}^{(j)}$ after training, the actual ground truth values of $\rho^{(j)}$ and $c^{(j)}$ for each system can be accurately predicted via linear regression. 94

Figure 34 Example reconstructions of several subjects from the experimental fMRI dataset (Sect. A.3.2) using a hierarchical shPLRNN with $N_{feat} = 15$, $L = 300$, and trained using GTF with $\alpha = 0.1$ and $T_{seq} = 72$. 95

Figure 35 **a**: Cosine similarity matrix based on the average similarity of the feature vectors across ten training runs for four example subjects. **b**: Extracted similarities and cluster labels reflect visual differences in the recorded BOLD signals. 96

Figure 36 **a**: Plateau effect observed for the number of total subregions traversed for different reconstructed DS and different PLRNN dimensions (Lorenz-63: $M = 10$, Lorenz-96: $M = 30$, bursting neuron: $M = 25$, Duffing: $M = 20$, Rössler: $M = 8$) **b**: Scaling of the total number of linear subregions of a PLRNN, given its latent dimension M, versus the number of subregions inhabited by trained Lorenz-63 systems with different dimensions. While the number of total subregions scales exponentially with M, the used linear subregions increase much more slowly. **c**: Number of boundary crossings per time step for a trajectory with $100,000$ time steps. For the Lorenz-63 and Rössler systems, the models do not cross any boundaries on most time steps, illustrating that the dynamics are highly linearized. **d**: Cumulative frequencies of the individual subregions for trained systems. **e**: Reconstructed attractors for the Lorenz-63 and bursting neuron model, colored with respect to the linear subregions corresponding to each observation. 97

Figure 37 **a**: Connectome of transitions between linear sub-regions, sorted by their relative frequency, for a PLRNN trained on a Lorenz-63 system, and resulting graph structure visualized using the spectral layout in `networkx`. The resulting graph shadows the layout of the real Lorenz-63 system, with the most frequented subregion (label 0) at the center of the intersection between the left and right lobe. **b**: Connectome for a reconstructed bursting neuron model, using the same layout. The graph for this system mimics the cyclic nature of the system and lacks a similarly dominant and interconnected sub-graph. 99

Figure 38 **a**: Illustration of geometry-based pruning. The top row shows a ground truth and reconstructed Lorenz-63 attractor (blue) and a successful reconstruction (red). The bottom row illustrates reconstructions where a single weight was removed with varying influence on attractor geometry. The shift in difference in geometric importance score D_{stsp} does not necessarily relate to the absolute magnitude of the pruned parameter indicated below. **b**: Weight parameters with large ($\Delta D_{stsp} > 0.1$) vs. low ($\Delta D_{stsp} \leqslant 0.1$) impact on geometrical reconstruction quality only feature a small difference in absolute magnitude. This observation illustrates why magnitude can not be meaningfully leveraged for pruning DSR models. Taken from [149]. Created by Christoph Hemmer. 102

Figure 39 Approach for translating graph-topological properties of trained networks into a general scheme to be used as topological prior. From [149]. 104

Figure 40 Example graph topologies with network sparsity of 85%. Hubs with $\geqslant 6$ connections are marked in red. From [149]. 104

Figure 41 **a**: The Trust Game (TG) setup, where participants, presented with an image of one of four virtual trustees, are given 50 fictitious monetary units for investment. The invested sum is then tripled and passed on to the trustee. Participants subsequently receive feedback detailing their investment, the trustee's repayment, and the retained amount. **b**: Example trial. **c**: The RNN model training process, mimicking the TG setup. The RNN, after receiving fairness and expression inputs, forecasts future investments. Based on its forecasts, the RNN is updated with data on the repayment and new balance. 107

Figure 42 Model Prediction: **a**: Mean linear prediction error (MLE; left) and correlation (right) between predicted and actual investments in the test set across all 32 participants, including a comparison of selected models' performance against random investment choices marked by the red bar. Red dots are selected models, error bars are SEM. **b**: Observed and model-predicted investments for a subset of participants for selected models. **c**: Observed vs. predicted average investment behavior based on the trustee for both social (left) and non-social (right) conditions. 110

Figure 43 **a** Left: Cumulative density for average correlations between entropy and first principal component (PC; light shade), and investment and second PC (dark shade) across latent trajectories of the true experimental simulation for all 32 participants. Right: Observation model entropy over investments, averaged over actual interactions in social (dark shade) and non-social (light shade) conditions. **b**: State space projection onto the first two PCs for two example participants, structured by an entropy gradient along the first PC (left) and investment gradient along the second PC (right). 111

Figure 44 **a**: Cosine similarity matrix of input vectors (left) and resulting clustering dendrogram (right) for social (top) and non-social (bottom) conditions. **b**: Average input strength comparison for strong (++ and –) versus weak (+ and -) expression cues in social and non-social conditions. Bottom right: Average input strength across all cues between social and non-social conditions is higher in the social condition. 112

Figure 45 Simulated behavior of an example participant in response to the presentation of an unfair trustee at 4 varying intensity levels ($c = 0, c = 0.6, c = 1.0, c = 1.4$). The investment in the absence of any input is at the maximum investment value of 50. As input intensity is up-regulated, the investment behavior (and corresponding system dynamics) exhibits a qualitative change, also referred to as a bifurcation. The participant first enters an exploratory state ($c = 0.6$), whose precise nature depends on the input strength as well. When the unfair trustee is displayed at full intensity ($c = 1.4$), the investment reaches a minimum (reflecting an unwillingness to cooperate with the unfair trustee). 113

Figure 46 **a**: Interactions of models with four example trustees with different interaction styles. **b**: Example predicted strategies (y-axis) over time (x-axis) for models with successful investment strategies for the antagonistic trustee (AT, the cooperative trustee (CT), the trust-building trustee (BT), and the uncooperative trustee (UT). **c**: Clustering based on average relative gain extracted from different trustees. Low values indicate low ranks, i.e., good performance and high extracted gains. In the off-diagonal panels, the x and y axis denote the ranks of the 32 participants for the respective trustee condition. 114

Figure 47 **a**: Comparison of rank for the CT vs. AT for subjects in the social vs. non-social group. **b**: Extracted gain across social vs. non-social group, averaged across all four agents. 115

Figure 48 Estimates of the first and second moment from the Adam optimizer for different parameters of the shPLRNN during training on the chaotic Lorenz-63 system. **a**: Without TF, even for a moderate sequence length (here $T = 30$), gradients retain high variance, making successful training impossible. **b**: With GTF ($\alpha = 0.2$), gradient moments decrease over time after a short initial period with high variance, stabilizing training even for much longer sequences (here $T = 200$). 129

Figure 49 Example reconstructions using SINDy, comparing reconstructions from a trajectory solving (red) and not solving (black) an algebraic equation in the parameters. From [139]. 132

Figure 50 For the GT shown top-left, providing SINDy only with a polynomial library leads to an incorrect solution for the VF (top-right), while providing the correct library including both trigonometric and polynomial functions (and mixing terms) leads to the correctly inferred VF (bottom-left). Finally, even with the correct library, providing as data the curve solving an algebraic trajectory leads to an incorrectly inferred VF (bottom-right). From [139]. 132

Figure 51 SINDy needs the proper function library to correctly infer a system across the whole state space (left). If the 3rd order term present in the Duffing equations is lacking (right), the inferred VF may only be locally correct (or not at all for more complex systems). From [139]. 133
Figure 52 The basic principle of Reservoir Computing. 133
Figure 53 The basic principle of Neural ODEs. 134
Figure 54 Short-term predictions (**a**) and long-term spatiotemporal behavior (**b**) for our models and N-ODEs for the multi-scale Lorenz-96 system, introduced in [68] to assess forecasting performance of different DSR approaches. Taken from [152]. 142
Figure 55 5-step-ahead predictions (yellow) of DSR algorithms on EEG time series (blue). While all methods provide reasonable short-term forecasts, reflecting the optimization criterion, most fail to produce non-trivial limiting dynamics (Fig. 23). Taken from [152]. 143
Figure 56 Different approaches for symbolizing the same underlying dynamical system. In preliminary experiments, MTF managed to approximately reconstruct the underlying DS from all four symbolic representations. 144
Figure 57 Example reconstructions of spike trains and spike statistics on the training set (see Methods A.3.2). Test set reconstructions and further statistics are in Figure 32. Taken from [53]. 144

LIST OF TABLES

Table 1 Classification of benchmark systems in the DS literature. 27
Table 2 Performance comparison of encoder and RNN models trained using MTF on multimodal data from the chaotic Lorenz system, using the performance metrics introduced in Sect. 4.1. Taken from [53]. 58
Table 3 Comparison of the dendPLRNN trained by VI or BPTT+TF, RC [303], LSTM-MSM [408], SINDy [59] and Neural ODE ([70]) on 4 DS benchmarks (top) and 3 challenging data situations (bottom). Values are mean ± SEM. Based on [54]. 77
Table 4 Comparisons of SOTA DSR algorithms on two challenging experimental datasets (Sect. A.3), adapted from [152]. Values are median ± median absolute deviation over 20 runs. 'dim' refers to the model's dynamical variables, while $|\boldsymbol{\theta}|$ are number of *trainable* parameters. Based on [152]. 79

Table 5 Comparison of dendPLRNN trained by MTF, by a multimodal SVAE based on [214], a VAE-TF approach similar to MTF except that all data modalities were 'Gaussianized' (GVAE-TF), BPTT-TF as in [54] using Gaussianized data, and a multiple-shooting (MS) approach. Training was performed on multivariate normal, ordinal, and count data produced by the chaotic Lorenz system, Rössler system, and Lewis-Glass model. Values are mean ± SEM, averaged over 15 trained models. X = value cannot be computed for this model (e.g., because resp. decoder model is not present). SCC (Spearman cross-correlation), OACF (ordinal autocorrelation function), and CACF (count autocorrelation function) all refer to mean-squared-errors (MSEs) between ground truth and generated correlation functions. Bold numbers indicate top performance within ±1 SEM. Taken from [53]. 81

Table 6 Comparison among multi-modal reconstruction methods for experimental fMRI+behavioral data. For each subject and training method, medians across 15 trained models were first obtained for each measure, which were then averaged across 20 subjects (± SEM). SEM = standard error of the mean. X = value not accessible for this method. The abbreviations are the same as in Table 5. Taken from [53]. 91

Table 7 Comparisons of SOTA DSR methods on the Lorenz-63 and Lorenz-96 system. Note that our training methods perform approximately on par with SINDy, which features a strong inductive bias in favor of reconstructing the two benchmark systems since these are low-order polynomials which the provided library included in these experiments (see Sect. A.2.1). Taken from [152]. 141

Table 8 Comparison of dendPLRNN trained by MTF with the MVAE [214], and an MS approach, on 8 ordinal observations with seven ordered categories, produced by the chaotic Lorenz system, Rössler system, and Lewis-Glass model, and on a symbolic representation of the chaotic Lorenz system. Values are mean ± SEM, averaged over 15 trained models. Taken from [53]. 143

BIBLIOGRAPHY

[1] Unai Fischer Abaigar. "Modeling Ordinal Mobile Data with Sequential Variational Autoencoders." Master's Thesis. Heidelberg, Germany: University of Heidelberg, 2022.

[2] Henry D. I. Abarbanel, Reggie Brown, John J. Sidorowich, and Lev Sh. Tsimring. "The analysis of observed chaotic data in physical systems." In: *Reviews of Modern Physics* 65.4 (Oct. 1993). Publisher: American Physical Society, pp. 1331–1392. DOI: 10.1103/RevModPhys.65.1331.

[3] Henry Abarbanel. *Predicting the future: completing models of observed complex systems*. Springer, 2013.

[4] Jeffrey Aguilar et al. "A review on locomotion robophysics: the study of movement at the intersection of robotics, soft matter and dynamical systems." eng. In: *Reports on Progress in Physics. Physical Society (Great Britain)* 79.11 (Nov. 2016), p. 110001. ISSN: 1361-6633. DOI: 10.1088/0034-4885/79/11/110001.

[5] Chaitanya Ahuja, Louis Philippe Morency, et al. "Multimodal machine learning: A survey and taxonomy." In: *IEEE Transactions of Pattern Analysis and Machine Intelligence* (2017), pp. 1–20.

[6] Adedotun Akintayo and Soumik Sarkar. "Hierarchical symbolic dynamic filtering of streaming non-stationary time series data." In: *Signal Processing* 151 (Oct. 2018), pp. 76–88. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2018.04.025.

[7] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks." In: *Reviews of Modern Physics* 74.1 (2002), pp. 47–97. DOI: 10.1103/revmodphys.74.47.

[8] Igor L. Aleiner, Lara Faoro, and Lev B. Ioffe. "Microscopic model of quantum butterfly effect: Out-of-time-order correlators and traveling combustion waves." In: *Annals of Physics* 375 (Dec. 2016), pp. 378–406. ISSN: 0003-4916. DOI: 10.1016/j.aop.2016.09.006.

[9] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Red. by Thomas F. Banchoff, Keith Devlin, Gaston Gonnet, Jerrold Marsden, and Stan Wagon. Textbooks in Mathematical Sciences. New York, NY: Springer, 1996. ISBN: 978-0-387-94677-1 978-0-387-22492-3. DOI: 10.1007/b97589.

[10] Joao Lucas de Sousa Almeida, Pedro Rocha, Allan Carvalho, and Alberto Costa Nogueira Junior. "A coupled Variational Encoder-Decoder - DeepONet surrogate model for the Rayleigh-Bénard convection problem." en-US. In: *AAAI 2023*. 2023.

[11] Victor M. Martinez Alvarez, Rareş Roşca, and Cristian G. Fălcuţescu. *DyNODE: Neural Ordinary Differential Equations for Dynamics Modeling in Continuous Control*. Sept. 2020. DOI: 10.48550/arXiv.2009.04278.

[12] Azmeraw T. Amare, Klaus Oliver Schubert, and Bernhard T. Baune. "Pharmacogenomics in the treatment of mood disorders: Strategies and Opportunities for personalized psychiatry." en. In: *EPMA Journal* 8.3 (Sept. 2017), pp. 211–227. ISSN: 1878-5085. DOI: 10.1007/s13167-017-0112-8.

[13] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. "Noise Estimation Using Density Estimation for Self-Supervised Multimodal Learning." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (May 2021), pp. 6644–6652. DOI: 10.1609/aaai.v35i8.16822.

[14] Abdul Fatir Ansari et al. *Chronos: Learning the Language of Time Series*. arXiv:2403.07815 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2403.07815.

[15] Luigi Antelmi, Nicholas Ayache, Philippe Robert, and Marco Lorenzi. "Multi-channel Stochastic Variational Inference for the Joint Analysis of Heterogeneous Biomedical Data in Alzheimer's Disease." In: *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*. Cham: Springer International Publishing, 2018, pp. 15–23. ISBN: 978-3-030-02628-8.

[16] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. "Black box variational inference for state space models." In: (2015). URL: http://arxiv.org/abs/1511.07367.

[17] Ludwig Arnold. *Random Dynamical Systems*. Springer Monographs in Mathematics. Berlin, Heidelberg: Springer, 1998. ISBN: 978-3-642-08355-6 978-3-662-12878-7. DOI: 10.1007/978-3-662-12878-7.

[18] Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael W. Mahoney. "Forecasting Sequential Data using Consistent Koopman Autoencoders." In: *Proceedings of the 37th International Conference on Machine Learning*. 2020. URL: http://arxiv.org/abs/2003.02236.

[19] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization." In: (July 2016). URL: http://arxiv.org/abs/1607.06450.

[20] Junwen Bai, Weiran Wang, and Carla P Gomes. "Contrastively Disentangled Sequential Variational Autoencoder." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 10105–10118.

[21] Joseph Bakarji, Kathleen Champion, J. Nathan Kutz, and Steven L. Brunton. "Discovering governing equations from partial measurements with deep delay autoencoders." In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 479.2276 (Aug. 2023). Publisher: Royal Society, p. 20230422. DOI: 10.1098/rspa.2023.0422.

[22] Bart Bakker and Tom Heskes. "Learning and approximate inference in dynamic hierarchical models." In: *Computational Statistics & Data Analysis* 52.2 (Oct. 2007), pp. 821–839. ISSN: 0167-9473. DOI: 10.1016/j.csda.2007.01.001.

[23] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal Machine Learning: A Survey and Taxonomy." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (Feb. 2019), pp. 423–443. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2798607.

[24] Albert-Laszlo Barabási and Reka Albert. "Emergence of Scaling in Random Networks." In: *Science* 286.5439 (1999), pp. 509–512. DOI: 10.1126/science.286.5439.509. eprint: http://www.sciencemag.org/cgi/reprint/286/5439/509.pdf.

[25] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. "Automatic Differentiation in Machine Learning: a Survey." In: *Journal of Machine Learning Research* 18.153 (2018), pp. 1–43. ISSN: 1533-7928. URL: http://jmlr.org/papers/v18/17-468.html.

[26] Justin Bayer, Maximilian Soelch, Atanas Mirchev, Baris Kayalibay, and Patrick van der Smagt. "Mind the Gap when Conditioning Amortised Inference in Sequential Latent-Variable Models." In: *arXiv:2101.07046 [cs, stat]* (2021). arXiv: 2101.07046.

[27] Philip Becker-Ehmck, Jan Peters, and Patrick Van Der Smagt. "Switching Linear Dynamics for Variational Bayes Filtering." en. In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, May 2019, pp. 553–562.

[28] John M. Beggs and Dietmar Plenz. "Neuronal avalanches in neocortical circuits." eng. In: *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience* 23.35 (Dec. 2003), pp. 11167–11177. ISSN: 1529-2401. DOI: 10.1523/JNEUROSCI.23-35-11167.2003.

[29] Mikhail Belkin. "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation." en. In: *Acta Numerica* 30 (May 2021). Publisher: Cambridge University Press, pp. 203–248. ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S0962492921000039.

[30] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. "Reconciling modern machine-learning practice and the classical bias–variance trade-off." In: *Proceedings of the National Academy of Sciences* 116.32 (Aug. 2019). Publisher: Proceedings of the National Academy of Sciences, pp. 15849–15854. DOI: 10.1073/pnas.1903070116.

[31] Mikhail Belkin and Partha Niyogi. "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering." In: *Advances in Neural Information Processing Systems*. Vol. 14. MIT Press, 2001. URL: https://proceedings.neurips.cc/paper_files/paper/2001/hash/f106b7f99d2cb30c3db1c3cc0fde9ccb-Abstract.html.

[32] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. "Parameterized quantum circuits as machine learning models." en. In: *Quantum Science and Technology* 4.4 (Nov. 2019). Publisher: IOP Publishing, p. 043001. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab4eb5.

[33] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult." In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166. ISSN: 1045-9227. DOI: 10.1109/72.279181.

[34] Konstantinos Benidis et al. "Deep Learning for Time Series Forecasting: Tutorial and Literature Survey." In: *ACM Computing Surveys* 55.6 (July 31, 2023), pp. 1–36. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3533382.

[35] Leonard Bereska and Efstratios Gavves. "Continual learning of dynamical systems with competitive federated reservoir computing." In: *Conference on Lifelong Learning Agents*. PMLR, 2022, pp. 335–350. URL: https://proceedings.mlr.press/v199/bereska22a.html.

[36] Joyce Berg, John Dickhaut, and Kevin McCabe. "Trust, Reciprocity, and Social History." In: *Games and Economic Behavior* 10.1 (July 1995), pp. 122–142. ISSN: 0899-8256. DOI: 10.1006/game.1995.1027.

[37] Nils Bertschinger and Thomas Natschläger. "Real-time computation at the edge of chaos in recurrent neural networks." eng. In: *Neural Computation* 16.7 (July 2004), pp. 1413–1436. ISSN: 0899-7667. DOI: 10.1162/089976604323057443.

[38] M. J. Betancourt and Mark Girolami. *Hamiltonian Monte Carlo for Hierarchical Models*. arXiv:1312.0906 [stat]. Dec. 2013. DOI: 10.48550/arXiv.1312.0906.

[39] Nikhil Bhagwat, Joseph D. Viviano, Aristotle N. Voineskos, M. Mallar Chakravarty, and Alzheimer's Disease Neuroimaging Initiative. "Modeling and prediction of clinical symptom trajectories in Alzheimer's disease using longitudinal data." In: *PLOS Computational Biology* 14.9 (Sept. 14, 2018), e1006376. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006376.

[40] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. "Quantum machine learning." en. In: *Nature* 549.7671 (Sept. 2017). Publisher: Nature Publishing Group, pp. 195–202. ISSN: 1476-4687. DOI: 10.1038/nature23474.

[41] Marcel Binz and Eric Schulz. "Using cognitive psychology to understand GPT-3." In: *Proceedings of the National Academy of Sciences* 120.6 (Feb. 2023). Publisher: Proceedings of the National Academy of Sciences, e2218523120. DOI: 10.1073/pnas.2218523120.

[42] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. "What is the state of neural network pruning?" In: *Proceedings of machine learning and systems* 2 (2020), pp. 129–146.

[43] François Blanchard. *Topological chaos: what may this mean ?* en. arXiv:0805.0232 [math]. May 2008. URL: http://arxiv.org/abs/0805.0232.

[44] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. "Variational Inference: A Review for Statisticians." In: *Journal of the American Statistical Association* 112.518 (2017), 859–877. ISSN: 1537-274X. DOI: 10.1080/01621459.2017.1285773.

[45] Stefan Bloemheuvel, Jurgen van den Hoogen, Dario Jozinović, Alberto Michelini, and Martin Atzmueller. "Graph Neural Networks for Multivariate Time Series Regression with Application to Seismic Data." In: *International Journal of Data Science and Analytics* (Aug. 30, 2022). ISSN: 2364-415X, 2364-4168. DOI: 10.1007/s41060-022-00349-6.

[46] H. G. Bock and K. J. Plitt. "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems*." In: *IFAC Proceedings Volumes*. 9th IFAC World Congress: A Bridge Between Control Science and Technology 17.2 (July 1, 1984), pp. 1603–1608. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)61205-9.

[47] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. "Sliced and radon wasserstein barycenters of measures." In: *Journal of Mathematical Imaging and Vision* 51 (2015), pp. 22–45.

[48] G. E. P. Box and D. R. Cox. "An Analysis of Transformations." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 26.2 (1964). Publisher: Royal Statistical Society, Wiley, pp. 211–252. ISSN: 0035-9246. URL: https://www.jstor.org/stable/2984418.

[49] George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Google-Books-ID: 1WVHAAAAMAAJ. Holden-Day, 1976. 616 pp. ISBN: 978-0-8162-1104-3.

[50] Edward S. Boyden. "Optogenetics and the future of neuroscience." en. In: *Nature Neuroscience* 18.9 (Sept. 2015). Publisher: Nature Publishing Group, pp. 1200–1201. ISSN: 1546-1726. DOI: 10.1038/nn.4094.

[51] Leo Breiman. "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)." In: *Statistical Science* 16.3 (Aug. 2001). Publisher: Institute of Mathematical Statistics, pp. 199–231. ISSN: 0883-4237, 2168-8745. DOI: 10.1214/ss/1009213726.

[52] Manuel Brenner, Christoph Hemmer, and Daniel Durstewitz. "Almost-Linear RNNs Yield Highly Interpretable Symbolic Codes in Dynamical Systems Reconstruction." Preprint. 2024.

[53] Manuel Brenner, Florian Hess, Georgia Koppe, and Daniel Durstewitz. *Integrating Multimodal Data for Joint Generative Modeling of Complex Dynamics*. Published at AAAI 2023 MLmDS workshop as "Multimodal Teacher Forcing for Reconstructing Nonlinear Dynamical Systems". Accepted at ICML 2024. DOI: 10.48550/arXiv.2212.07892.

[54] 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*. ISSN: 2640-3498. PMLR, June 2022, pp. 2292–2320. URL: https://proceedings.mlr.press/v162/brenner22a.html.

[55] Manuel Brenner, Christoph Korn, Daniela Mier, Stephanie N.L. Schmidt, Stefanie Lis, Peter Kirsch, Daniel Durstewitz, and Georgia Koppe. "Creating digital twins of social interaction partners through deep dynamical systems learning." Preprint. 2024.

[56] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, and J. Nathan Kutz. "Chaos as an intermittently forced linear system." en. In: *Nature Communications* 8.1 (May 2017). Publisher: Nature Publishing Group, p. 19. ISSN: 2041-1723. DOI: 10.1038/s41467-017-00030-8.

[57] Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. *Modern Koopman Theory for Dynamical Systems*. arXiv:2102.12086 [cs, eess, math]. Oct. 2021. DOI: 10.48550/arXiv.2102.12086.

[58] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[59] 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 USA* 113.15 (2016), pp. 3932–3937. ISSN: 0027-8424. DOI: 10.1073/pnas.1517384113.

[60] Rebekka Burkholz, Nilanjana Laha, Rajarshi Mukherjee, and Alkis Gotovos. *On the Existence of Universal Lottery Tickets*. arXiv:2111.11146 [cs, stat]. Mar. 2022. DOI: 10.48550/arXiv.2111.11146.

[61] Thomas Bury, Daniel Dylewsky, Chris Bauch, Madhur Anand, Leon Glass, Alvin Shrier, and Gil Bub. *Predicting discrete-time bifurcations with deep learning*. Mar. 2023.

[62] Gyorgy Buzsaki. *Rhythms of the Brain*. en. Google-Books-ID: ldz58irprjYC. Oxford University Press, Aug. 2006. ISBN: 978-0-19-804125-2.

[63] Thomas L. Carroll and Louis M. Pecora. "Network Structure Effects in Reservoir Computers." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.8 (Aug. 2019), p. 083130. ISSN: 1054-1500, 1089-7682. DOI: 10.1063/1.5097686.

[64] Martin Casdagli. "Nonlinear prediction of chaotic time series." In: *Physica D: Nonlinear Phenomena* 35.3 (May 1989), pp. 335–356. ISSN: 0167-2789. DOI: 10.1016/0167-2789(89)90074-2.

[65] Fausto Cavalli and Ahmad Naimzada. "Complex dynamics and multistability with increasing rationality in market games." In: *Chaos, Solitons & Fractals* 93 (2016), pp. 151–161. ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2016.10.014.

[66] Bo Chang, Minmin Chen, Eldad Haber, and Ed H. Chi. "AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks." In: *International Conference on Learning Representations* (Feb. 25, 2019). arXiv: 1902.09689.

[67] Luke J. Chang, Bradley B. Doll, Mascha van 't Wout, Michael J. Frank, and Alan G. Sanfey. "Seeing is believing: trustworthiness as a dynamic belief." eng. In: *Cognitive Psychology* 61.2 (Sept. 2010), pp. 87–105. ISSN: 1095-5623. DOI: 10.1016/j.cogpsych.2010.03.001.

[68] Ashesh Chattopadhyay, Pedram Hassanzadeh, and Devika Subramanian. "Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network." In: *Nonlinear Processes in Geophysics* 27.3 (2020), pp. 373–389.

[69] Adam M. Chekroud et al. "The promise of machine learning in predicting treatment outcomes in psychiatry." en. In: *World Psychiatry* 20.2 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wps.20882, pp. 154–170. ISSN: 2051-5545. DOI: 10.1002/wps.20882.

[70] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. "Neural Ordinary Differential Equations." In: *Advances in Neural Information Processing Systems 31*. 2018. URL: http://arxiv.org/abs/1806.07366.

[71] Shizhe Chen, Ali Shojaie, and Daniela M. Witten. "Network Reconstruction From High-Dimensional Ordinary Differential Equations." In: *Journal of the American Statistical Association* 112.520 (2017), pp. 1697–1707. ISSN: 0162-1459. DOI: 10.1080/01621459.2016.1229197.

[72] Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. "Symplectic Recurrent Neural Networks." en. In: *Proceedings of the 8th International Conference on Learning Representations*. 2020.

[73] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[74] Krzysztof Ciesielski. "The Poincaré-Bendixson Theorem: from Poincaré to the XXIst century." en. In: *Central European Journal of Mathematics* 10.6 (Dec. 2012), pp. 2110–2128. ISSN: 1644-3616. DOI: 10.2478/s11533-012-0110-y.

[75] Jorge Corral-Acero et al. "The 'Digital Twin' to enable the vision of precision cardiology." In: *European Heart Journal* 41.48 (Mar. 2020), pp. 4556–4564. ISSN: 0195-668X. DOI: 10.1093/eurheartj/ehaa159.

[76] Alexandre Cortiella, Kwang-Chun Park, and Alireza Doostan. "Sparse identification of nonlinear dynamical systems via reweighted 1-regularized least squares." In: *Computer Methods in Applied Mechanics and Engineering* 376 (Apr. 2021), p. 113620. ISSN: 0045-7825. DOI: 10.1016/j.cma.2020.113620.

[77] Antonio C. Costa, Tosif Ahamed, and Greg J. Stephens. "Adaptive, locally linear models of complex dynamics." In: *Proceedings of the National Academy of Sciences* 116.5 (Jan. 2019). Publisher: Proceedings of the National Academy of Sciences, pp. 1501–1510. DOI: 10.1073/pnas.1813476116.

[78] Zhicheng Cui, Wenlin Chen, and Yixin Chen. "Multi-Scale Convolutional Neural Networks for Time Series Classification." In: *Computing Research Repository* abs/1603.06995 (2016). arXiv: 1603.06995.

[79] Matthew Dale, Simon O'Keefe, Angelika Sebald, Susan Stepney, and Martin A. Trefzer. "Reservoir computing quality: connectivity and topology." en. In: *Natural Computing* 20.2 (June 2021), pp. 205–216. ISSN: 1572-9796. DOI: 10.1007/s11047-020-09823-1.

[80] Alex Damian, Tengyu Ma, and Jason D. Lee. *Label Noise SGD Provably Prefers Flat Global Minimizers*. en. June 2021. URL: https://arxiv.org/abs/2106.06530v2.

[81] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. *A decoder-only foundation model for time-series forecasting*. Feb. 2024. DOI: 10.48550/arXiv.2310.10688.

[82] George Datseris. "DynamicalSystems.jl: A Julia software library for chaos and nonlinear dynamics." In: *Journal of Open Source Software* 3.23 (2018), p. 598. DOI: 10.21105/joss.00598.

[83] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. "The Helmholtz machine." eng. In: *Neural Computation* 7.5 (Sept. 1995), pp. 889–904. ISSN: 0899-7667. DOI: 10.1162/neco.1995.7.5.889.

[84] Shaan Desai, Marios Mattheakis, Hayden Joy, Pavlos Protopapas, and Stephen Roberts. *One-Shot Transfer Learning of Physics-Informed Neural Networks*. arXiv:2110.11286 [physics]. July 2022. DOI: 10.48550/arXiv.2110.11286.

[85] Amir Dezfouli, Richard Morris, Fabio T Ramos, Peter Dayan, and Bernard Balleine. "Integrated accounts of behavioral and neuroimaging data using flexible recurrent neural network models." In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[86] Sander Dieleman et al. *Continuous diffusion for categorical data*. Dec. 2022. DOI: 10.48550/arXiv.2211.15089.

[87] J. R. Dormand and P. J. Prince. "A family of embedded Runge-Kutta formulae." In: *Journal of Computational and Applied Mathematics* 6.1 (Mar. 1980), pp. 19–26. ISSN: 0377-0427. DOI: 10.1016/0771-050X(80)90013-3.

[88] Kenji Doya. "Bifurcations in the learning of recurrent neural networks." en. In: *Proceedings of the 1992 IEEE International Symposium on Circuits and Systems*. 1992. ISBN: 978-0-7803-0593-9. DOI: 10.1109/ISCAS.1992.230622.

[89] Jianghua Duan, Yongsheng Ou, Jianbing Hu, Zhiyang Wang, Shaokun Jin, and Chao Xu. "Fast and Stable Learning of Dynamical Systems Based on Extreme Learning Machine." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.6 (June 2019), pp. 1175–1185. ISSN: 2168-2232. DOI: 10.1109/TSMC.2017.2705279.

[90] Jinqiao Duan. *An Introduction to Stochastic Dynamics*. en. Google-Books-ID: jYWEBwAAQBAJ. Cambridge University Press, Apr. 2015. ISBN: 978-1-107-07539-9.

[91] Jorge Duarte, Cristina Januário, Nuno Martins, and Josep Sardanyés. "Quantifying chaos for ecological stoichiometry." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 20.3 (Sept. 2010), p. 033105.

[92] Veronika Dubinkina, Yulia Fridman, Parth Pratim Pandey, and Sergei Maslov. "Multistability and regime shifts in microbial communities explained by competition for essential nutrients." In: *eLife* 8 (2019), e49720. ISSN: 2050-084X. DOI: 10.7554/eLife.49720.

[93] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. "Augmented Neural ODEs." In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[94] Daniel Durstewitz. "Implications of synaptic biophysics for recurrent network dynamics and active memory." en. In: *Neural Networks* 22.8 (2009), pp. 1189–1200. ISSN: 08936080. DOI: 10.1016/j.neunet.2009.07.016.

[95] Daniel Durstewitz. "A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements." eng. In: *PLoS Comput. Biol.* 13.6 (2017), e1005542. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005542.

[96] Daniel Durstewitz, Quentin J. M. Huys, and Georgia Koppe. "Psychiatric Illnesses as Disorders of Network Dynamics." eng. In: *Biological Psychiatry. Cognitive Neuroscience and Neuroimaging* 6.9 (Sept. 2021), pp. 865–876. ISSN: 2451-9030. DOI: 10.1016/j.bpsc.2020.01.001.

[97] Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. "Reconstructing computational system dynamics from neural data with recurrent neural networks." eng. In: *Nature Reviews. Neuroscience* 24.11 (Nov. 2023), pp. 693–710. ISSN: 1471-0048. DOI: 10.1038/s41583-023-00740-7.

[98] Daniel Durstewitz, Georgia Koppe, and Hazem Toutounji. "Computational models as statistical tools." In: *Current Opinion in Behavioral Sciences*. Computational modeling 11 (Oct. 2016), pp. 93–99. ISSN: 2352-1546. DOI: 10.1016/j.cobeha.2016.07.004.

[99] Daniel Durstewitz, Jeremy K. Seamans, and Terrence J. Sejnowski. "Neurocomputational models of working memory." en. In: *Nature Neuroscience* 3.11 (Nov. 2000). Number: 11 Publisher: Nature Publishing Group, pp. 1184–1191. ISSN: 1546-1726. DOI: 10.1038/81460.

[100] 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." eng. In: *Neuron* 66.3 (May 2010), pp. 438–448. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2010.03.029.

[101] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin. "Pruning and regularization in reservoir computing." In: *Neurocomputing*. Advances in Machine Learning and Computational Intelligence 72.7 (Mar. 2009), pp. 1534–1546. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2008.12.020.

[102] J.-P. Eckmann and D. Ruelle. "Ergodic theory of chaos and strange attractors." en. In: *The Theory of Chaotic Attractors*. Ed. by Brian R. Hunt, Tien-Yien Li, Judy A. Kennedy, and Helena E. Nusse. New York, NY: Springer, 2004, pp. 273–312. ISBN: 978-0-387-21830-4. DOI: 10.1007/978-0-387-21830-4_17.

[103] Maria K. Eckstein, Christopher Summerfield, Nathaniel D. Daw, and Kevin J. Miller. *Predictive and Interpretable: Combining Artificial Neural Networks and Classic Cognitive Models to Understand Human Learning and Decision Making*. en. May 2023. DOI: 10.1101/2023.05.17.541226.

[104] Lukas Eisenmann, Zahra Monfared, Niclas Göring, and Daniel Durstewitz. "Bifurcations and loss jumps in RNN training." In: *Advances in Neural Information Processing Systems* 36 (2023).

[105] Jeffrey L. Elman. "Language as a dynamical system." In: *Mind as motion: Explorations in the dynamics of cognition*. Cambridge, MA, US: The MIT Press, 1995, pp. 195–225. ISBN: 978-0-262-16150-3.

[106] N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. "Lipschitz Recurrent Neural Networks." In: *International Conference on Learning Representations* (Apr. 23, 2021). arXiv: 2006.12070.

[107] D. Reid Evans and Diane Larsen-Freeman. "Bifurcations and the Emergence of L2 Syntactic Structures in a Complex Dynamic System." English. In: *Frontiers in Psychology* 11 (Oct. 2020). Publisher: Frontiers. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2020.574603.

[108] Marisa Faggini. "Chaotic time series analysis in economics: Balance and perspectives." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 24.4 (2014). Publisher: American Institute of Physics, p. 042101.

[109] Mohammad Farazmand and Themistoklis P. Sapsis. *Extreme Events: Mechanisms and Prediction*. Mar. 2018. DOI: 10.48550/arXiv.1803.06277.

[110] Janik Fechtelpeter, Christian Rauschenberg, Hamidreza Jamalabadi, Benjamin Boecking, Therese van Amelsvoort, Ulrich Reininghaus, Daniel Durstewitz, and Georgia Koppe. "A control theoretic approach to evaluate and inform ecological momentary interventions." en-us. In: (Feb. 2024). Publisher: OSF. DOI: 10.31234/osf.io/97teh.

[111] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. "Testing the manifold hypothesis." en. In: *Journal of the American Mathematical Society* 29.4 (Oct. 2016), pp. 983–1049. ISSN: 0894-0347, 1088-6834. DOI: 10.1090/jams/852.

[112] Jonathan Frankle and Michael Carbin. "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks." In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rJl-b3RcF7.

[113] Nele Franzen, Meike Hagenhoff, Nina Baer, Ariane Schmidt, Daniela Mier, Gebhard Sammer, Bernd Gallhofer, Peter Kirsch, and Stefanie Lis. "Superior 'theory of mind' in borderline personality disorder: An analysis of interaction behavior in a virtual trust game." In: *Psychiatry Research* 187.1-2 (2011), pp. 224–233. ISSN: 1872-7123. DOI: 10.1016/j.psychres.2010.11.012.

[114] Karl Friston, Rosalyn J. Moran, Yukie Nagai, Tadahiro Taniguchi, Hiroaki Gomi, and Josh Tenenbaum. "World model learning and inference." In: *Neural Networks* 144 (Dec. 2021), pp. 573–590. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2021.09.011.

[115] Kunihiko Fukushima. "Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements." In: *IEEE Transactions on Systems Science and Cybernetics* 5.4 (Oct. 1969), pp. 322–333. ISSN: 2168-2887. DOI: 10.1109/TSSC.1969.300225.

[116] Ken-ichi Funahashi. "On the approximate realization of continuous mappings by neural networks." In: *Neural Networks* (1989). DOI: 10.1016/0893-6080(89)90003-8.

[117] Christophe Gauld and Damien Depannemaecker. "Dynamical systems in computational psychiatry: A toy-model to apprehend the dynamics of psychiatric symptoms." English. In: *Frontiers in Psychology* 14 (Feb. 2023). Publisher: Frontiers. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2023.1099257.

[118] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis, Third Edition*. en. CRC Press, Nov. 2013. ISBN: 978-1-4398-4095-5.

[119] Nicholas Geneva and Nicholas Zabaras. "Transformers for modeling physical systems." In: *Neural Networks* 146 (Feb. 2022), pp. 272–289. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2021.11.022.

[120] Amin Ghadami and Bogdan I. Epureanu. "Forecasting critical points and post-critical limit cycles in nonlinear oscillatory systems using pre-critical transient responses." In: *International Journal of Non-Linear Mechanics* 101 (May 2018), pp. 146–156. ISSN: 0020-7462. DOI: 10.1016/j.ijnonlinmec.2018.02.008.

[121] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. *From Variational to Deterministic Autoencoders*. en. arXiv:1903.12436 [cs, stat]. May 2020. URL: http://arxiv.org/abs/1903.12436.

[122] Sanjib Ghosh, Kohei Nakajima, Tanjung Krisnanda, Keisuke Fujii, and Timothy C. H. Liew. "Quantum Neuromorphic Computing with Reservoir Computing Networks." en. In: *Advanced Quantum Technologies* 4.9 (2021), p. 2100053. ISSN: 2511-9044. DOI: 10.1002/qute.202100053.

[123] Dimitrios Giannakis, Abbas Ourmazd, Philipp Pfeffer, Jörg Schumacher, and Joanna Slawinska. "Embedding classical dynamics in a quantum computer." In: *Physical Review A* 105.5 (May 2022). Publisher: American Physical Society, p. 052404. DOI: 10.1103/PhysRevA.105.052404.

[124] J. Gidi, B. Candia, A. D. Muñoz-Moller, A. Rojas, L. Pereira, M. Muñoz, L. Zambrano, and A. Delgado. "Stochastic optimization algorithms for quantum applications." In: *Physical Review A* 108.3 (Sept. 2023). Publisher: American Physical Society, p. 032409. DOI: 10.1103/PhysRevA.108.032409.

[125] William Gilpin. "Chaos as an interpretable benchmark for forecasting and data-driven modelling." In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2022. URL: https://openreview.net/forum?id=enYjtbjYJrf.

[126] William Gilpin. "Model scale versus domain knowledge in statistical forecasting of chaotic systems." In: *Physical Review Research* 5.4 (Dec. 2023). Publisher: American Physical Society, p. 043252. DOI: 10.1103/PhysRevResearch.5.043252.

[127] William Gilpin. "Generative learning for nonlinear dynamics." en. In: *Nature Reviews Physics* 6.3 (Mar. 2024). Publisher: Nature Publishing Group, pp. 194–206. ISSN: 2522-5820. DOI: 10.1038/s42254-024-00688-2.

[128] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. "Dynamical Variational Autoencoders: A Comprehensive Review." In: *Foundations and Trends® in Machine Learning* 15.1 (2021), pp. 1–175. ISSN: 1935-8237, 1935-8245. arXiv: 2008.12595.

[129] James Gleick. *Chaos: Making a New Science*. en. Open Road Media, Mar. 2011. ISBN: 978-1-4532-1047-5.

[130] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." en. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. ISSN: 1938-7228. JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 249–256.

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

[132] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets." In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014.

[133] Andres D. Grosmark, J. Long, and György Buzsáki. *Recordings from hippocampal area CA1, PRE, during and POST novel spatial learning.* Mar. 2016. DOI: http://dx.doi.org/10.6080/K0862DC5.

[134] E. P. Gross. "Structure of a quantized vortex in boson systems." en. In: *Il Nuovo Cimento (1955-1965)* 20.3 (May 1961), pp. 454–477. ISSN: 1827-6121. DOI: 10.1007/BF02731494.

[135] Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. arXiv:2312.00752 [cs]. Dec. 2023. DOI: 10.48550/arXiv.2312.00752.

[136] Albert Gu, Karan Goel, and Christopher Ré. *Efficiently Modeling Long Sequences with Structured State Spaces*. arXiv:2111.00396 [cs]. Aug. 2022. DOI: 10.48550/arXiv.2111.00396.

[137] Yali Guo, Han Zhang, Liang Wang, Huawei Fan, Jinghua Xiao, and Xingang Wang. "Transfer learning of chaotic systems." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.1 (Jan. 2021), p. 011104. ISSN: 1054-1500. DOI: 10.1063/5.0033870.

[138] Stephen Guth and Themistoklis P. Sapsis. "Machine Learning Predictors of Extreme Events Occurring in Complex Dynamical Systems." en. In: *Entropy* 21.10 (Oct. 2019). Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, p. 925. ISSN: 1099-4300. DOI: 10.3390/e21100925.

[139] Niclas Göring, Florian Hess, Manuel Brenner, Zahra Monfared, and Daniel Durstewitz. *Out-of-Domain Generalization in Dynamical Systems Reconstruction*. Accepted at ICML 2024. DOI: 10.48550/arXiv.2402.18377.

[140] Jutho Haegeman, J. Ignacio Cirac, Tobias J. Osborne, Iztok Pižorn, Henri Verschelde, and Frank Verstraete. "Time-Dependent Variational Principle for Quantum Lattices." In: *Physical Review Letters* 107.7 (Aug. 2011). Publisher: American Physical Society, p. 070601. DOI: 10.1103/PhysRevLett.107.070601.

[141] Richard Hahnloser and H. Sebastian Seung. "Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks." In: *Advances in Neural Information Processing Systems*. Vol. 13. MIT Press, 2000.

[142] Hamel. "Georg Duffing, Ingenieur: Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung. Sammlung Vieweg. Heft 41/42, Braunschweig 1918. VI+134 S." en. In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 1.1 (1921), pp. 72–73. DOI: 10.1002/zamm.19210010109.

[143] Song Han, Huizi Mao, and William J. Dally. *Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding*. en. arXiv:1510.00149 [cs]. Feb. 2016. URL: http://arxiv.org/abs/1510.00149.

[144] Xiaoying Han and Peter E. Kloeden. *Random Ordinary Differential Equations and Their Numerical Solution*. Vol. 85. Probability Theory and Stochastic Modelling. Singapore: Springer, 2017. ISBN: 978-981-10-6264-3 978-981-10-6265-0. DOI: 10.1007/978-981-10-6265-0.

[145] Xinyu Han, Yi Zhao, and Michael Small. "A tighter generalization bound for reservoir computing." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 32.4 (Apr. 2022), p. 043115. ISSN: 1054-1500. DOI: 10.1063/5.0082258.

[146] Joshua Hanson and Maxim Raginsky. "Universal Simulation of Stable Dynamical Systems by Recurrent Neural Nets." In: *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. Vol. 120. Proceedings of Machine Learning Research. PMLR, 2020, pp. 384–392. URL: https://proceedings.mlr.press/v120/hanson20a.html.

[147] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[148] Niklas Heim, Václav Šmídl, and Tomáš Pevný. *Rodent: Relevance determination in differential equations*. Mar. 2020. DOI: 10.48550/arXiv.1912.00656.

[149] Christoph Hemmer, Manuel Brenner, Florian Hess, and Daniel Durstewitz. "Optimal Network Topologies for Dynamical Systems Reconstruction." Accepted at ICML 2024.

[150] Douglas Hendry and Adrian E. Feiguin. "Machine learning approach to dynamical properties of quantum many-body systems." In: *Physical Review B* 100.24 (Dec. 2019). Publisher: American Physical Society, p. 245123. DOI: 10.1103/PhysRevB.100.245123.

[151] John R. Hershey and Peder A. Olsen. "Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models." In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07* 4 (2007), pp. IV–317–IV–320.

[152] 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*. ISSN: 2640-3498. PMLR, July 2023, pp. 13017–13049. URL: https://proceedings.mlr.press/v202/hess23a.html.

[153] Selina Hiller, Christian Rauschenberg, Christian Götzl, Janik Fechtelpeter, Georgia Koppe, Daniel Durstewitz, Ulrich Reininghaus, and Silvia Krumm. "Gemeinsam gestalten?! Wie junge Menschen, Praxis und Wissenschaft bei der Entwicklung einer Smartphone-App zusammenfinden. Das Reallabor AI4U (engl. „Artificial Intelligence for Youth") – Zwischenergebnisse eines Partizipationsprojekts zur Entwicklung einer auf Künstlicher Intelligenz basierten Smartphone-App zur Gesundheitsförderung junger Menschen." In: *unsere jugend* 75 (Jan. 2023), pp. 77–91. DOI: 10.2378/uj2023.art11d.

[154] Geoffrey E Hinton. "Training products of experts by minimizing contrastive divergence." In: *Neural computation* 14.8 (2002), pp. 1771–1800.

[155] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory." In: *Neural Comput.* 9.8 (1997), 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[156] Holger F. Hofmann. "On the fundamental role of dynamics in quantum physics." en. In: *The European Physical Journal D* 70.5 (May 2016), p. 118. ISSN: 1434-6079. DOI: 10.1140/epjd/e2016-70086-8.

[157] J 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 (Apr. 1982), pp. 2554–2558. ISSN: 0027-8424. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238/.

[158] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." en. In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366. ISSN: 0893-6080.

[159] Honor Hsin, Menachem Fromer, Bret Peterson, Collin Walter, Mathias Fleck, Andrew Campbell, Paul Varghese, and Robert Califf. "Transforming Psychiatry into Data-Driven Medicine with Digital Measurement Tools." en. In: *npj Digital Medicine* 1.1 (Aug. 2018). Publisher: Nature Publishing Group, pp. 1–4. ISSN: 2398-6352. DOI: 10.1038/s41746-018-0046-0.

[160] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv:2106.09685 [cs]. Oct. 2021. DOI: 10.48550/arXiv.2106.09685.

[161] Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V. Albert, and John Preskill. "Provably efficient machine learning for quantum many-body problems." en. In: *Science* 377.6613 (Sept. 2022). arXiv:2106.12627 [quant-ph], eabk3333. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.abk3333.

[162] Alfred Hubler. "Is symbolic dynamics the most efficient data compression tool for chaotic time series?" In: *Complexity* 17.3 (2012), pp. 5–7.

[163] Andreas Hula, P. Read Montague, and Peter Dayan. "Monte Carlo Planning Method Estimates Planning Horizons during Interactive Social Exchange." en. In: *PLOS Computational Biology* 11.6 (June 2015). Publisher: Public Library of Science, e1004254. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1004254.

[164] Andreas Hula, Iris Vilares, Terry Lohrenz, Peter Dayan, and P. Read Montague. "A model of risk and mental state shifts during social interaction." en. In: *PLOS Computational Biology* 14.2 (Feb. 2018), e1005935. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005935.

[165] Quentin J. M. Huys, Tiago V. Maia, and Michael J. Frank. "Computational psychiatry as a bridge from neuroscience to clinical applications." eng. In: *Nature Neuroscience* 19.3 (Mar. 2016), pp. 404–413. ISSN: 1546-1726. DOI: 10.1038/nn.4238.

[166] Michael Häusser, Nelson Spruston, and Greg J. Stuart. "Diversity and Dynamics of Dendritic Signaling." en. In: *Science* 290.5492 (2000), pp. 739–744. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.290.5492.739.

[167] Masanobu Inubushi and Susumu Goto. "Transfer learning for nonlinear dynamics and its application to fluid turbulence." In: *Physical Review E* 102.4 (Oct. 2020). Publisher: American Physical Society, p. 043301. DOI: 10.1103/PhysRevE.102.043301.

[168] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. "Deep learning for time series classification: a review." en. In: *Data Mining and Knowledge Discovery* 33.4 (July 2019), pp. 917–963. ISSN: 1573-756X. DOI: 10.1007/s10618-019-00619-1.

[169] Anthony R. Ives and Vasilis Dakos. "Detecting dynamical changes in nonlinear time series using locally linear state-space models." en. In: *Ecosphere* 3.6 (2012). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1890/ES11-00347.1, art58. ISSN: 2150-8925. DOI: 10.1890/ES11-00347.1.

[170] Eugene M. Izhikevich. *Dynamical systems in neuroscience: the geometry of excitability and bursting*. en. Computational neuroscience. OCLC: ocm65400606. Cambridge, Mass: MIT Press, 2007. ISBN: 978-0-262-09043-8.

[171] Mozes Jacobs, Bingni W. Brunton, Steven L. Brunton, J. Nathan Kutz, and Ryan V. Raut. *HyperSINDy: Deep Generative Modeling of Nonlinear Stochastic Governing Equations*. arXiv:2310.04832 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2310.04832.

[172] Nicholas C. Jacobson and Yeon Joo Chung. "Passive Sensing of Prediction of Moment-To-Moment Depressed Mood among Undergraduates with Clinical Levels of Depression Sample Using Smartphones." In: *Sensors (Basel, Switzerland)* 20.12 (June 2020), p. 3572. ISSN: 1424-8220. DOI: 10.3390/s20123572.

[173] Nicholas C. Jacobson, Damien Lekkas, Raphael Huang, and Natalie Thomas. "Deep Learning Paired with Wearable Passive Sensing Data Predicts Deterioration in Anxiety Disorder Symptoms across 17–18 Years." In: *Journal of affective disorders* 282 (Mar. 2021), pp. 104–111. ISSN: 0165-0327. DOI: 10.1016/j.jad.2020.12.086.

[174] Herbert Jaeger and Harald Haas. "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication." en. In: *Science* 304.5667 (2004), pp. 78–80. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1091277.

[175] Tim Januschowski, Yuyang Wang, Kari Torkkola, Timo Erkkilä, Hilaf Hasson, and Jan Gasthaus. "Forecasting with trees." In: *International Journal of Forecasting*. Special Issue: M5 competition 38.4 (Oct. 2022), pp. 1473–1481. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2021.10.004.

[176] Bin Jiang and Christophe Claramunt. "Topological analysis of urban street networks." In: *Environment and Planning B: Planning and design* 31.1 (2004), pp. 151–162.

[177] Matt Johnson, Zico Kolter, and David Duvenaud. *Deep Implicit Layers - Neural ODEs, Deep Equilibirum Models, and Beyond*. en. 2020. URL: http://implicit-layers-tutorial.org/.

[178] Michael I Jordan. "Attractor dynamics and parallelism in a connectionist sequential machine." In: *Artificial neural networks: concept learning*. IEEE Press, 1990, pp. 112–127.

[179] M. K. Joshi, F. Kranzl, A. Schuckert, I. Lovas, C. Maier, R. Blatt, M. Knap, and C. F. Roos. "Observing emergent hydrodynamics in a long-range quantum magnet." In: *Science* 376.6594 (May 2022). Publisher: American Association for the Advancement of Science, pp. 720–724. DOI: 10.1126/science.abk2400.

[180] Laércio Oliveira Junior, Florian Stelzer, and Liang Zhao. "Clustered Echo State Networks for Signal Observation and Frequency Filtering." en. In: *Anais do Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*. ISSN: 2763-8944. SBC, Oct. 2020, pp. 25–32. DOI: 10.5753/kdmile.2020.11955.

[181] Anil Kag, Ziming Zhang, and Venkatesh Saligrama. "RNNs Incrementally Evolving on an Equilibrium Manifold: A Panacea for Vanishing and Exploding Gradients?" In: *International Conference on Learning Representations* (2020), p. 24.

[182] Kadierdan Kaheman, Steven L. Brunton, and J. Nathan Kutz. "Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data." en. In: *Machine Learning: Science and Technology* 3.1 (Mar. 2022). Publisher: IOP Publishing, p. 015031. ISSN: 2632-2153. DOI: 10.1088/2632-2153/ac567a.

[183] E. Kaiser, J. N. Kutz, and S. L. Brunton. "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit." In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2219 (Nov. 2018). Publisher: Royal Society, p. 20180335. DOI: 10.1098/rspa.2018.0335.

[184] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems." In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[185] Sebastian Kaltenbach and Phaedon-Stelios Koutsourelakis. "Incorporating physical constraints in a deep probabilistic machine learning framework for coarse-graining dynamical systems." In: *Journal of Computational Physics* 419 (Oct. 2020), p. 109673. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109673.

[186] Léandre Kamdjeu Kengne, Justin R. Mboupda Pone, and Hilaire B. Fotsin. "On the dynamics of chaotic circuits based on memristive diode-bridge with variable symmetry: A case study." In: *Chaos, Solitons & Fractals* 145 (2021), p. 110795.

[187] Holger Kantz and Thomas Schreiber. *Nonlinear time series analysis*. Vol. 7. Cambridge university press, 2004.

[188] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. "Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data." In: *Proceedings of the 5th International Conference on Learning Representations*. 2017. URL: http://arxiv.org/abs/1605.06432.

[189] Daniel Karlsson and Olle Svanström. "Modelling Dynamical Systems Using Neural Ordinary Differential Equations." eng. In: (2019). URL: https://hdl.handle.net/20.500.12380/256887.

[190] K. Kashinath et al. "Physics-informed machine learning: case studies for weather and climate modelling." In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (Feb. 2021). Publisher: Royal Society, p. 20200093. DOI: 10.1098/rsta.2020.0093.

[191] Anatole Katok, A. B. Katok, and Boris Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*. en. Google-Books-ID: 9nL7ZX8Djp4C. Cambridge University Press, 1995. ISBN: 978-0-521-57557-7.

[192] Kavli Institute for Systems Neuroscience. *Liam Paninski: Accelerating the experimental-analysis-theory cycle*. Feb. 2022. URL: https://www.youtube.com/watch?v=VsIAMEkAYO0.

[193] Giancarlo Kerg, Kyle Goyette, Maximilian Puelma Touzel, Gauthier Gidel, Eugene Vorontsov, Yoshua Bengio, and Guillaume Lajoie. "Non-normal Recurrent Neural Network (nnRNN): learning long time dependencies while improving expressivity with transient dynamics." In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (2019), p. 11.

[194] Farshad Khadivar, Ilaria Lauzana, and Aude Billard. "Learning dynamical systems with bifurcations." In: *Robotics and Autonomous Systems* 136 (Feb. 2021), p. 103700. ISSN: 0921-8890. DOI: 10.1016/j.robot.2020.103700.

[195] Saeed Khaki, Hieu Pham, Ye Han, Andy Kuhl, Wade Kent, and Lizhi Wang. "Convolutional Neural Networks for Image-Based Corn Kernel Detection and Counting." en. In: *Sensors* 20.9 (Jan. 2020). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 2721. ISSN: 1424-8220. DOI: 10.3390/s20092721.

[196] Vedika Khemani, David A. Huse, and Adam Nahum. "Velocity-dependent Lyapunov exponents in many-body quantum, semiclassical, and classical chaos." In: *Physical Review B* 98.14 (Oct. 2018). Publisher: American Physical Society, p. 144304. DOI: 10.1103/PhysRevB.98.144304.

[197] Mikail Khona and Ila R. Fiete. "Attractor and integrator networks in the brain." en. In: *Nature Reviews Neuroscience* 23.12 (Dec. 2022). Number: 12 Publisher: Nature Publishing Group, pp. 744–766. ISSN: 1471-0048. DOI: 10.1038/s41583-022-00642-0.

[198] Samuel Kim, Peter Y. Lu, Srijon Mukherjee, Michael Gilbert, Li Jing, Vladimir Čeperić, and Marin Soljačić. "Integration of Neural Network-Based Symbolic Regression in Deep Learning for Scientific Discovery." In: *IEEE Transactions on Neural Networks and Learning Systems* 32.9 (Sept. 2021), pp. 4166–4177. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2020.3017010.

[199] Suyong Kim, Weiqi Ji, Sili Deng, Yingbo Ma, and Christopher Rackauckas. "Stiff Neural Ordinary Differential Equations." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.9 (Sept. 2021). arXiv:2103.15341 [cs, math], p. 093122. ISSN: 1054-1500, 1089-7682. DOI: 10.1063/5.0060697.

[200] Masahiro Kimura and Ryohei Nakano. "Learning dynamical systems by recurrent neural networks from orbits." In: *Neural Networks* 11.9 (1998), pp. 1589–1599.

[201] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *Proceedings of the 3rd International Conference on Learning Representations*. 2015. URL: http://arxiv.org/abs/1412.6980.

[202] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes." en. In: *Proceedings of the 2nd International Conference on Learning Representations*. 2014. URL: http://arxiv.org/abs/1312.6114.

[203] Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. *Generalizing to New Physical Systems via Context-Informed Dynamics Model*. arXiv:2202.01889 [cs, stat]. June 2022. DOI: 10.48550/arXiv.2202.01889.

[204] Joon-Hyuk Ko, Hankyul Koh, Nojun Park, and Wonho Jhe. *Homotopy-based training of NeuralODEs for accurate dynamics discovery*. arXiv:2210.01407 [physics]. May 2023. URL: http://arxiv.org/abs/2210.01407.

[205] Christof Koch. *Biophysics of computation: information processing in single neurons*. Oxford university press, 2004.

[206] Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. "Machine learning–accelerated computational fluid dynamics." In: *Proceedings of the National Academy of Sciences* 118.21 (May 2021). Publisher: Proceedings of the National Academy of Sciences, e2101784118. DOI: 10.1073/pnas.2101784118.

[207] Doogesh Kodi Ramanah, Radosław Wojtak, Zoe Ansari, Christa Gall, and Jens Hjorth. "Dynamical mass inference of galaxy clusters with neural flows." In: *Monthly Notices of the Royal Astronomical Society* 499.2 (Oct. 2020), pp. 1985–1997. ISSN: 0035-8711. DOI: 10.1093/mnras/staa2886.

[208] B. O. Koopman and J. v. Neumann. "Dynamical Systems of Continuous Spectra." In: *Proceedings of the National Academy of Sciences* 18.3 (Mar. 1932). Publisher: Proceedings of the National Academy of Sciences, pp. 255–263. DOI: 10.1073/pnas.18.3.255.

[209] Georgia Koppe, Harald Gruppe, Gebhard Sammer, Bernd Gallhofer, Peter Kirsch, and Stefanie Lis. "Temporal unpredictability of a stimulus sequence affects brain activation differently depending on cognitive task demands." In: *NeuroImage* 101 (Nov. 1, 2014), pp. 236–244. ISSN: 1095-9572. DOI: 10.1016/j.neuroimage.2014.07.008.

[210] Georgia Koppe, Sinan Guloksuz, Ulrich Reininghaus, and Daniel Durstewitz. "Recurrent Neural Networks in Mobile Sampling and Intervention." eng. In: *Schizophrenia Bulletin* 45.2 (Mar. 2019), pp. 272–276. ISSN: 1745-1701. DOI: 10.1093/schbul/sby171.

[211] Georgia Koppe, Andreas Meyer-Lindenberg, and Daniel Durstewitz. "Deep learning for small and big data in psychiatry." eng. In: *Neuropsychopharmacology: Official Publication of the American College of Neuropsychopharmacology* 46.1 (Jan. 2021), pp. 176–190. ISSN: 1740-634X. DOI: 10.1038/s41386-020-0767-z.

[212] Georgia Koppe, Hazem Toutounji, Peter Kirsch, Stefanie Lis, and Daniel Durstewitz. "Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI." en. In: *PLOS Computational Biology* 15.8 (2019), e1007263. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1007263.

[213] Irena Koprinska, Dengsong Wu, and Zheng Wang. "Convolutional Neural Networks for Energy Time Series Forecasting." In: *2018 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407. July 2018, pp. 1–8. DOI: 10.1109/IJCNN.2018.8489399.

[214] 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. URL: https://proceedings.mlr.press/v162/kramer22a.html.

[215] Kai-Hauke Krämer, George Datseris, Jürgen Kurths, Istvan Z Kiss, Jorge L Ocampo-Espindola, and Norbert Marwan. "A unified and automated approach to attractor reconstruction." In: *New Journal of Physics* 23.3 (2021), p. 033017.

[216] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012.

[217] Dmitry Krotov and John Hopfield. *Large Associative Memory Problem in Neurobiology and Machine Learning*. arXiv:2008.06996 [cond-mat, q-bio, stat]. Apr. 2021. DOI: 10.48550/arXiv.2008.06996.

[218] Nathan Kutz. *The Future of Governing Equations*. 2023. URL: https://amath.washington.edu/news/2022/02/25/nathan-kutz-future-governing-equations.

[219] Y. A. Kuznetsov and R. J. Sacker. "Neimark-Sacker bifurcation." In: *Scholarpedia* 3.5 (2008). revision #91556, p. 1845. DOI: 10.4249/scholarpedia.1845.

[220] Yuri A. Kuznetsov. *Elements of Applied Bifurcation Theory (2nd Ed.)* Berlin, Heidelberg: Springer-Verlag, 1998. ISBN: 0387983821.

[221] William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. *Contemporary Symbolic Regression Methods and their Relative Performance*. arXiv:2107.14351 [cs]. July 2021. DOI: 10.48550/arXiv.2107.14351.

[222] Remi Lam et al. "Learning skillful medium-range global weather forecasting." In: *Science* 382.6677 (Dec. 2023). Publisher: American Association for the Advancement of Science, pp. 1416–1421. DOI: 10.1126/science.adi2336.

[223] Diane Lambert. "Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing." In: *Technometrics* 34.1 (1992). Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality], pp. 1–14. ISSN: 0040-1706. DOI: 10.2307/1269547.

[224] Itamar Daniel Landau and Haim Sompolinsky. "Coherent chaos in a recurrent neural network with structured connectivity." en. In: *PLoS Comput Biol* 14.12 (2018), e1006309. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006309.

[225] Peter E. Latham and Sheila Nirenberg. "Synergy, Redundancy, and Independence in Population Codes, Revisited." In: *The Journal of Neuroscience* 25.21 (May 2005), pp. 5195–5206. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5319-04.2005.

[226] David Layden, Guglielmo Mazzola, Ryan V. Mishmash, Mario Motta, Pawel Wocjan, Jin-Sung Kim, and Sarah Sheldon. "Quantum-enhanced Markov chain Monte Carlo." en. In: *Nature* 619.7969 (July 2023). Publisher: Nature Publishing Group, pp. 282–287. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06095-4.

[227] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D. Hager. *Temporal Convolutional Networks: A Unified Approach to Action Segmentation*. arXiv:1608.08242 [cs]. Aug. 2016. DOI: 10.48550/arXiv.1608.08242.

[228] Don S. Lemons and Anthony Gythiel. "Paul Langevin's 1908 paper "On the Theory of Brownian Motion" ["Sur la théorie du mouvement brownien," C. R. Acad. Sci. (Paris) 146, 530–533 (1908)]." In: *American Journal of Physics* 65.11 (Nov. 1997), pp. 1079–1081. ISSN: 0002-9505. DOI: 10.1119/1.18725.

[229] John E. Lewis and Leon Glass. "Nonlinear Dynamics and Symbolic Dynamics of Neural Networks." In: *Neural Computation* 4.5 (1992), pp. 621–642. ISSN: 0899-7667. DOI: 10.1162/neco.1992.4.5.621.

[230] Boning Li and Akane Sano. "Extraction and Interpretation of Deep Autoencoder-based Temporal Features from Wearables for Forecasting Personalized Mood, Health, and Stress." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.2 (June 2020), 49:1–49:26. DOI: 10.1145/3397318.

[231] Fanjun Li, Ying Li, and Xiaohong Wang. "Echo State Network with Hub Property." en. In: *Proceedings of 2019 Chinese Intelligent Automation Conference*. Ed. by Zhidong Deng. Lecture Notes in Electrical Engineering. Singapore: Springer, 2020, pp. 537–544. ISBN: 978-981-329-050-1. DOI: 10.1007/978-981-32-9050-1_61.

[232] Yikuan Li, Shishir Rao, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Gholamreza Salimi-Khorshidi, Mohammad Mamouei, Thomas Lukasiewicz, and Kazem Rahimi. "Deep Bayesian Gaussian processes for uncertainty estimation in electronic health records." en. In: *Scientific Reports* 11.1 (Oct. 2021). Publisher: Nature Publishing Group, p. 20685. ISSN: 2045-2322. DOI: 10.1038/s41598-021-00144-6.

[233] Yuhong Li, Tianle Cai, Yi Zhang, Deming Chen, and Debadeepta Dey. *What Makes Convolutional Models Great on Long Sequence Modeling?* Oct. 17, 2022. URL: http://arxiv.org/abs/2210.09298.

[234] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. *Fourier Neural Operator with Learned Deformations for PDEs on General Geometries*. July 11, 2022. arXiv: 2207.05209[cs,math]. URL: http://arxiv.org/abs/2207.05209.

[235] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. "Fourier Neural Operator for Parametric Partial Differential Equations." en. In: *Proceedings of the 9th International Conference on Learning Representations*. 2021.

[236] Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. "Learning Chaotic Dynamics in Dissipative Systems." en. In: *Advances in Neural Information Processing Systems* 35 (Dec. 2022), pp. 16768–16781.

[237] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. *Physics-Informed Neural Operator for Learning Partial Differential Equations*. Nov. 13, 2022. arXiv: 2111.03794[cs,math]. URL: http://arxiv.org/abs/2111.03794.

[238] Muxuan Liang, Zhizhong Li, Ting Chen, and Jianyang Zeng. "Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach." In: *IEEE/ACM transactions on computational biology and bioinformatics* 12.4 (Aug. 2015), pp. 928–937. ISSN: 1557-9964. DOI: 10.1109/TCBB.2014.2377729.

[239] Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. "Foundations and recent trends in multimodal machine learning: Principles, challenges, and open questions." In: *arXiv preprint arXiv:2209.03430* (2022).

[240] Torrin M. Liddell and John K. Kruschke. "Analyzing ordinal data with metric models: What could possibly go wrong?" In: *Journal of Experimental Social Psychology* 79 (Nov. 1, 2018), pp. 328–348. ISSN: 0022-1031. DOI: 10.1016/j.jesp.2018.08.009.

[241] R. Likert. "A technique for the measurement of attitudes." In: *Archives of Psychology* 22 140 (1932), pp. 55–55.

[242] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting." In: *International Journal of Forecasting* 37.4 (Oct. 2021), pp. 1748–1764. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2021.03.012.

[243] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge: Cambridge University Press, 1995. DOI: 10.1017/CBO9780511626302.

[244] Scott W. Linderman, Andrew C. Miller, Ryan P. Adams, David M. Blei, Liam Paninski, and Matthew J. Johnson. "Recurrent switching linear dynamical systems." In: *arXiv:1610.08466 [stat]* (Oct. 2016).

[245] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. "Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems." en. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2017, pp. 914–922. URL: https://proceedings.mlr.press/v54/linderman17a.html.

[246] Jana Lipkova et al. "Artificial intelligence for multimodal data integration in oncology." eng. In: *Cancer Cell* 40.10 (Oct. 2022), pp. 1095–1110. ISSN: 1878-3686. DOI: 10.1016/j.ccell.2022.09.012.

[247] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. *Flow Matching for Generative Modeling*. arXiv:2210.02747 [cs, stat]. Feb. 2023. DOI: 10.48550/arXiv.2210.02747.

[248] Stefanie Lis, Nina Baer, Nele Franzen, Meike Hagenhoff, Maika Gerlach, Georgia Koppe, Gebhard Sammer, Bernd Gallhofer, and Peter Kirsch. "Social Interaction Behavior in ADHD in Adults in a Virtual Trust Game." eng. In: *Journal of Attention Disorders* 20.4 (Apr. 2016), pp. 335–345. ISSN: 1557-1246. DOI: 10.1177/1087054713482581.

[249] 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." In: *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*. 2020.

[250] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. Aug. 17, 2021. DOI: 10.48550/arXiv.2103.14030. arXiv: 2103.14030[cs].

[251] Jean-Christophe Loiseau and Steven L. Brunton. "Constrained sparse Galerkin regression." en. In: *Journal of Fluid Mechanics* 838 (Mar. 2018). Publisher: Cambridge University Press, pp. 42–67. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/jfm.2017.823.

[252] Edward N Lorenz. "Deterministic nonperiodic flow." In: *Journal of atmospheric sciences* 20.2 (1963), pp. 130–141.

[253] Edward N Lorenz. "Predictability: A problem partly solved." In: *Proc. Seminar on predictability*. Vol. 1. 1996.

[254] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators." en. In: *Nature Machine Intelligence* 3.3 (Mar. 2021). Number: 3 Publisher: Nature Publishing Group, pp. 218–229. ISSN: 2522-5839. DOI: 10.1038/s42256-021-00302-5.

[255] Sirui Lu, Mari Carmen Bañuls, and J. Ignacio Cirac. "Algorithms for Quantum Simulation at Finite Energies." In: *PRX Quantum* 2.2 (May 2021). Publisher: American Physical Society, p. 020321. DOI: 10.1103/PRXQuantum.2.020321.

[256] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. "The expressive power of neural networks: a view from the width." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 6232–6240. ISBN: 978-1-5108-6096-4.

[257] Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. "Deep learning for universal linear embeddings of nonlinear dynamics." In: *Nat Commun* 9.1 (Dec. 2018). arXiv: 1712.09707, p. 4950. ISSN: 2041-1723. DOI: 10.1038/s41467-018-07210-0.

[258] Niru Maheswaranathan, Alex H. Williams, Matthew D. Golub, Surya Ganguli, and David Sussillo. "Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics." In: *Advances in neural information processing systems 32*. 2019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7416638/.

[259] Niru Maheswaranathan, Alex H. Williams, Matthew D. Golub, Surya Ganguli, and David Sussillo. "Universality and individuality in neural dynamics across large populations of recurrent networks." In: *Advances in Neural Information Processing Systems 32*. 2019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7416639/.

[260] Tiago V. Maia and Michael J. Frank. "From reinforcement learning models to psychiatric and neurological disorders." en. In: *Nature Neuroscience* 14.2 (Feb. 2011). Publisher: Nature Publishing Group, pp. 154–162. ISSN: 1546-1726. DOI: 10.1038/nn.2723.

[261] Nour Makke and Sanjay Chawla. "Interpretable scientific discovery with symbolic regression: a review." en. In: *Artificial Intelligence Review* 57.1 (Jan. 2024), p. 2. ISSN: 1573-7462. DOI: 10.1007/s10462-023-10622-0.

[262] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. "Proving the Lottery Ticket Hypothesis: Pruning is All You Need." en. In: *Proceedings of the 37th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, Nov. 2020, pp. 6682–6691. URL: https://proceedings.mlr.press/v119/malach20a.html.

[263] Benoit Mandelbrot and Richard L. Hudson. *The Misbehavior of Markets: A Fractal View of Financial Turbulence*. en. Google-Books-ID: GMKeUqufPQoC. Basic Books, Mar. 2007. ISBN: 978-0-465-00468-3.

[264] Sylvain Mangiarotti, Matthieu Peyre, Yixiao Zhang, Maciej Huc, Friederike Roger, and Yvonne Kerr. "Chaos theory applied to the outbreak of COVID-19: an ancillary approach to decision making in pandemic context." In: *Epidemiology and Infection* 148 (2020), e95.

[265] T. Matsumoto, H. Hamagishi, J. Sugi, and M. Saito. "From data-to dynamics: predicting chaotic time series by hierarchical Bayesian neural nets." In: *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*. Vol. 3. Anchorage, AK, USA: IEEE, 1998, pp. 2535–2540. ISBN: 978-0-7803-4859-2. DOI: 10.1109/IJCNN.1998.687261.

[266] T. Matsumoto, Y. Nakajima, M. Saito, J. Sugi, and H. Hamagishi. "Reconstructions and predictions of nonlinear dynamical systems: a hierarchical Bayesian approach." In: *IEEE Transactions on Signal Processing* 49.9 (Sept. 2001), pp. 2138–2155. ISSN: 1941-0476. DOI: 10.1109/78.942641.

[267] Peter McCullagh. "Regression Models for Ordinal Data." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 42.2 (1980), pp. 109–142. ISSN: 0035-9246. URL: https://www.jstor.org/stable/2984952.

[268] Michael McCullough, Konstantinos Sakellariou, Thomas Stemler, and Michael Small. "Regenerating time series from ordinal networks." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.3 (2017).

[269] Viraj Mehta, Ian Char, Willie Neiswanger, Youngseog Chung, Andrew Nelson, Mark Boyer, Egemen Kolemen, and Jeff Schneider. "Neural Dynamical Systems: Balancing Structure and Flexibility in Physical Prediction." In: *2021 60th IEEE Conference on Decision and Control (CDC)*. ISSN: 2576-2370. Dec. 2021, pp. 3735–3742. DOI: 10.1109/CDC45484.2021.9682807.

[270] Bartlett W. Mel. "Synaptic integration in an excitable dendritic tree." en. In: *Journal of Neurophysiology* 70.3 (1993), pp. 1086–1101. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.1993.70.3.1086.

[271] Bartlett W. Mel. "Information Processing in Dendritic Trees." In: *Neural Computation* 6.6 (1994), pp. 1031–1085. ISSN: 0899-7667. DOI: 10.1162/neco.1994.6.6.1031.

[272] Bartlett W. Mel. "Why Have Dendrites? A Computational Perspective." en_US. In: *Dendrites*. Oxford University Press, 1999. ISBN: 978-0-19-172420-6.

[273] Fanwang Meng, Michael Richer, Alireza Tehrani, Jonathan La, Taewon David Kim, Paul W. Ayers, and Farnaz Heidar-Zadeh. "Procrustes: A python library to find transformations that maximize the similarity between matrices." In: *Computer Physics Communications* 276.108334 (2022), pp. 1–37. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2022.108334.

[274] Daniel A. Messenger and David M. Bortz. "Weak SINDy: Galerkin-Based Data-Driven Model Selection." In: *Multiscale Modeling & Simulation* 19.3 (Jan. 2021). Publisher: Society for Industrial and Applied Mathematics, pp. 1474–1497. ISSN: 1540-3459. DOI: 10.1137/20M1343166.

[275] Stefan Mihai et al. "Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects." In: *IEEE Communications Surveys & Tutorials* 24.4 (2022). Conference Name: IEEE Communications Surveys & Tutorials, pp. 2255–2291. ISSN: 1553-877X. DOI: 10.1109/COMST.2022.3208773.

[276] Jonas Magdy Mikhaeil, Zahra Monfared, and Daniel Durstewitz. "On the difficulty of learning chaotic dynamics with RNNs." In: *Advances in Neural Information Processing Systems*. Oct. 31, 2022. URL: https://openreview.net/forum?id=-_AMpmyV0Ll.

[277] John A. Miller, Mohammed Aldosari, Farah Saeed, Nasid Habib Barna, Subas Rana, I. Budak Arpinar, and Ninghao Liu. *A Survey of Deep Learning and Foundation Models for Time Series Forecasting*. Jan. 2024. DOI: 10.48550/arXiv.2401.13912.

[278] John Milnor. "On the concept of attractor." en. In: *Communications in Mathematical Physics* 99.2 (June 1985), pp. 177–195. ISSN: 1432-0916. DOI: 10.1007/BF01212280.

[279] George S. Misyris, Andreas Venzke, and Spyros Chatzivasileiadis. *Physics-Informed Neural Networks for Power Systems*. arXiv:1911.03737 [cs, eess]. Jan. 2020. DOI: 10.48550/arXiv.1911.03737.

[280] Christoph Molnar. *Interpretable Machine Learning*. en. Google-Books-ID: jBm3DwAAQBAJ. Lulu.com, 2020. ISBN: 978-0-244-76852-2.

[281] Zahra Monfared and Daniel Durstewitz. "Existence of n-cycles and border-collision bifurcations in piecewise-linear continuous maps with applications to recurrent neural networks." en. In: *Nonlinear Dyn* 101.2 (2020), pp. 1037–1052. ISSN: 1573-269X. DOI: 10.1007/s11071-020-05841-x.

[282] Zahra Monfared and Daniel Durstewitz. "Transformation of ReLU-based recurrent neural networks from discrete-time to continuous-time." en. In: *Proceedings of the 37th International Conference on Machine Learning*. 2020.

[283] Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. *On the Number of Linear Regions of Deep Neural Networks*. arXiv:1402.1869 [cs, stat]. June 2014. DOI: 10.48550/arXiv.1402.1869.

[284] Edvard I. Moser, May-Britt Moser, and Bruce L. McNaughton. "Spatial representation in the hippocampal formation: a history." en. In: *Nature Neuroscience* 20.11 (Nov. 2017). Number: 11 Publisher: Nature Publishing Group, pp. 1448–1464. ISSN: 1546-1726. DOI: 10.1038/nn.4653.

[285] Ilan Naiman and Omri Azencot. "A Koopman Approach to Understanding Sequence Neural Models." en. In: *arXiv:2102.07824 [cs, math]* (Oct. 2021). URL: http://arxiv.org/abs/2102.07824.

[286] Aditya G. Nair, Steven L. Brunton, and Kunihiko Taira. "Networked-oscillator-based modeling and control of unsteady wake flows." In: *Physical Review E* 97.6 (June 2018). Publisher: American Physical Society, p. 063107. DOI: 10.1103/PhysRevE.97.063107.

[287] John B. Nezlek, Deborah S. Richardson, Laura R. Green, and Elizabeth C. Schatten-Jones. "Psychological well-being and day-to-day social interaction among older adults." In: *Personal Relationships* 9.1 (2002). Place: United Kingdom Publisher: Blackwell Publishing, pp. 57–71. ISSN: 1475-6811. DOI: 10.1111/1475-6811.00004.

[288] Calistus N. Ngonghala, Ulrike Feudel, and Kenneth Showalter. "Extreme multistability in a chemical model system." In: *Phys. Rev. E* 83 (5 2011), p. 056206. DOI: 10.1103/PhysRevE.83.056206.

[289] Steven A. Niederer, Michael S. Sacks, Mark Girolami, and Karen Willcox. "Scaling digital twins from the artisanal to the industrial." en. In: *Nature Computational Science* 1.5 (May 2021). Publisher: Nature Publishing Group, pp. 313–320. ISSN: 2662-8457. DOI: 10.1038/s43588-021-00072-5.

[290] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning Deconvolution Network for Semantic Segmentation." In: *2015 IEEE International Conference on Computer Vision (ICCV)*. ISSN: 2380-7504. Dec. 2015, pp. 1520–1528. DOI: 10.1109/ICCV.2015.178.

[291] J. O'Keefe and L. Nadel. *The Hippocampus as a Cognitive Map*. eng. Oxford, UK: Oxford University Press, 1978. URL: http://www.cognitivemap.net/.

[292] OpenAI et al. *GPT-4 Technical Report*. arXiv:2303.08774 [cs]. Dec. 2023. DOI: 10.48550/arXiv.2303.08774.

[293] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. "Logarithmic Pruning is All You Need." In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 2925–2934.

[294] George Osipenko. "Spectrum of a dynamical system and applied symbolic dynamics." In: *Journal of Mathematical Analysis and Applications* 252.2 (2000), pp. 587–616.

[295] George Osipenko and Stephen Campbell. "Applied symbolic dynamics: attractors and filtrations." en. In: *Discrete and Continuous Dynamical Systems* 5.1 (Sept. 1998). Publisher: Discrete and Continuous Dynamical Systems, pp. 43–60. ISSN: 1078-0947. DOI: 10.3934/dcds.1999.5.43.

[296] Samuel E. Otto and Clarence W. Rowley. *Linearly-Recurrent Autoencoder Networks for Learning Dynamics*. arXiv:1712.01378 [cs, math, stat]. Jan. 2019. DOI: 10.48550/arXiv.1712.01378.

[297] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. "Geometry from a Time Series." In: *Physical Review Letters* 45.9 (Sept. 1980), pp. 712–716. DOI: 10.1103/PhysRevLett.45.712.

[298] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning." In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (Oct. 2010), pp. 1345–1359. ISSN: 1558-2191. DOI: 10.1109/TKDE.2009.191.

[299] Chethan Pandarinath et al. "Inferring single-trial neural population dynamics using sequential auto-encoders." en. In: *Nature Methods* 15.10 (2018), pp. 805–815. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0109-9.

[300] L. Paninski and J. P. Cunningham. *Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience*. en. Pages: 196949 Section: New Results. Oct. 2017. DOI: 10.1101/196949.

[301] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." en. In: *Proceedings of the 30th International Conference on Machine Learning*. 2013. URL: http://proceedings.mlr.press/v28/pascanu13.html.

[302] Dhruvit Patel and Edward Ott. *Using Machine Learning to Anticipate Tipping Points and Extrapolate to Post-Tipping Dynamics of Non-Stationary Dynamical Systems*. en. arXiv:2207.00521 [physics]. July 2022. URL: http://arxiv.org/abs/2207.00521.

[303] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. "Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach." en. In: *Phys. Rev. Lett.* 120.2 (2018), p. 024102. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.024102.

[304] Jaideep Pathak, Zhixin Lu, Brian R. Hunt, Michelle Girvan, and Edward Ott. "Using Machine Learning to Replicate Chaotic Attractors and Calculate Lyapunov Exponents from Data." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.12 (Dec. 2017). arXiv: 1710.07313, p. 121102. ISSN: 1054-1500, 1089-7682. DOI: 10.1063/1.5010300.

[305] Jaideep Pathak et al. *FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators*. arXiv:2202.11214 [physics]. Feb. 2022. DOI: 10.48550/arXiv.2202.11214.

[306] Judea Pearl. "Bayesian networks." In: *The handbook of brain theory and neural networks*. Cambridge, MA, USA: MIT Press, Oct. 1998, pp. 149–153. ISBN: 978-0-262-51102-5.

[307] Barak A Pearlmutter. "Learning state space trajectories in recurrent neural networks." In: *Neural Computation* 1.2 (1989), pp. 263–269.

[308] Lawrence Perko. *Differential equations and dynamical systems*. en. 3rd ed. Texts in applied mathematics 7. New York: Springer, 2001. ISBN: 978-0-387-95116-4.

[309] Ya B. Pesin. "Characteristic Lyapunov Exponents and Smooth Ergodic Theory." en. In: *Russian Mathematical Surveys* 32.4 (Aug. 1977). Publisher: IOP Publishing, p. 55. ISSN: 0036-0279. DOI: 10.1070/RM1977v032n04ABEH001639.

[310] Fernando J Pineda. "Dynamics and architecture for neural computation." In: *Journal of Complexity* 4.3 (1988), pp. 216–245.

[311] Skyler Place et al. "Behavioral Indicators on a Mobile Sensing Platform Predict Clinically Validated Psychiatric Symptoms of Mood and Anxiety Disorders." EN. In: *Journal of Medical Internet Research* 19.3 (Mar. 2017), e6678. DOI: 10.2196/jmir.6678.

[312] Jason A. Platt, Stephen G. Penny, Timothy A. Smith, Tse-Chun Chen, and Henry D. I. Abarbanel. *A Systematic Exploration of Reservoir Computing for Forecasting Complex Spatiotemporal Dynamics*. arXiv:2201.08910 [cs]. Jan. 2022. DOI: 10.48550/arXiv.2201.08910.

[313] Jason A Platt, Stephen G Penny, Timothy A Smith, Tse-Chun Chen, and Henry DI Abarbanel. "Constraining chaos: Enforcing dynamical invariants in the training of reservoir computers." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 33.10 (2023).

[314] Ryan Po et al. *State of the Art on Diffusion Models for Visual Computing*. arXiv:2310.07204 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2310.07204.

[315] H. Poincaré. "Introduction." In: *Acta Mathematica* 13.1-2 (1890), p. 5. ISSN: 0001-5962.

[316] Panayiota Poirazi, Terrence Brannon, and Bartlett W Mel. "Pyramidal neuron as two-layer neural network." In: *Neuron* 37.6 (2003), pp. 989–999.

[317] Panayiota Poirazi and Athanasia Papoutsi. "Illuminating dendritic function with computational models." en. In: *Nat Rev Neurosci* 21.6 (2020), pp. 303–321. ISSN: 1471-003X, 1471-0048. DOI: 10.1038/s41583-020-0301-7.

[318] Di Qi and Andrew J. Majda. "Using machine learning to predict extreme events in complex systems." In: *Proceedings of the National Academy of Sciences* 117.1 (Jan. 2020). Publisher: Proceedings of the National Academy of Sciences, pp. 52–59. DOI: 10.1073/pnas.1917285117.

[319] Shaodi Qian, Chun-An Chou, and Jr-Shin Li. "Deep multi-modal learning for joint linear representation of nonlinear dynamical systems." In: *Scientific reports* 12.1 (2022), p. 12807.

[320] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision." en. In: *Proceedings of the 38th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2021, pp. 8748–8763. URL: https://proceedings.mlr.press/v139/radford21a.html.

[321] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." en. In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. ISSN: 00219991. DOI: 10.1016/j.jcp.2018.10.045.

[322] Maziar Raissi. "Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations." In: *Journal of Machine Learning Research* 19.25 (2018), pp. 1–24. URL: http://jmlr.org/papers/v19/18-046.html.

[323] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. *Zero-Shot Text-to-Image Generation*. arXiv:2102.12092 [cs]. Feb. 2021. DOI: 10.48550/arXiv.2102.12092.

[324] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. "Deep State Space Models for Time Series Forecasting." In: *Advances in Neural Information Processing Systems 31* (2018), pp. 7785–7794. URL: http://papers.nips.cc/paper/8004-deep-state-space-models-for-time-series-forecasting.pdf.

[325] Kashif Rasul et al. *Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting*. arXiv:2310.08278 [cs]. Feb. 2024. DOI: 10.48550/arXiv.2310.08278.

[326] P Read Montague, Terry Lohrenz, and Peter Dayan. "The three R's of trust." In: *Current Opinion in Behavioral Sciences*. Social behavior 3 (June 2015), pp. 102–106. ISSN: 2352-1546. DOI: 10.1016/j.cobeha.2015.02.009.

[327] Machel et al. Reid. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*. arXiv:2403.05530 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2403.05530.

[328] Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. "Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks." In: *Sensors* 19.14 (2019). ISSN: 1424-8220. DOI: 10.3390/s19143079.

[329] Rena L. Repetti. "Individual and common components of the social environment at work and psychological well-being." In: *Journal of Personality and Social Psychology* 52.4 (1987), pp. 710–720. ISSN: 1939-1315. DOI: 10.1037/0022-3514.52.4.710.

[330] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." In: *Proceedings of the 31st International Conference on Machine Learning*. 2014. URL: http://arxiv.org/abs/1401.4082.

[331] Danilo Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows." en. In: *Proceedings of the 32nd International Conference on Machine Learning*. 2015. URL: http://proceedings.mlr.press/v37/rezende15.html.

[332] Maria Isabel Ribeiro. "Kalman and Extended Kalman Filters: Concept, Derivation and Properties." en. In: (2004).

[333] Darius A. Rohani, Maria Faurholt-Jepsen, Lars Vedel Kessing, and Jakob E. Bardram. "Correlations Between Objective Behavioral Features Collected From Mobile and Wearable Devices and Depressive Mood Symptoms in Patients With Affective Disorders: Systematic Review." eng. In: *JMIR mHealth and uHealth* 6.8 (Aug. 2018), e165. ISSN: 2291-5222. DOI: 10.2196/mhealth.9691.

[334] Edmund T. Rolls. "A non-reward attractor theory of depression." eng. In: *Neuroscience and Biobehavioral Reviews* 68 (Sept. 2016), pp. 47–58. ISSN: 1873-7528. DOI: 10.1016/j.neubiorev.2016.05.007.

[335] Edmund T. Rolls, Marco Loh, and Gustavo Deco. "An attractor hypothesis of obsessive-compulsive disorder." eng. In: *The European Journal of Neuroscience* 28.4 (Aug. 2008), pp. 782–793. ISSN: 1460-9568. DOI: 10.1111/j.1460-9568.2008.06379.x.

[336] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. "Latent ODEs for Irregularly-Sampled Time Series." In: *arXiv:1907.03907 [cs, stat]* (July 2019). arXiv: 1907.03907. URL: http://arxiv.org/abs/1907.03907.

[337] Mikail Rubinov, Stuart A Knock, Cornelis J Stam, Sifis Micheloyannis, Anthony WF Harris, Leanne M Williams, and Michael Breakspear. "Small-world properties of nonlinear brain activity in schizophrenia." In: *Human brain mapping* 30.2 (2009), pp. 403–416.

[338] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. en. arXiv:1609.04747 [cs]. June 2017. URL: http://arxiv.org/abs/1609.04747.

[339] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." en. In: *Nature Machine Intelligence* 1.5 (May 2019). Publisher: Nature Publishing Group, pp. 206–215. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0048-x.

[340] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors." In: *Nature* 323.6088 (1986), pp. 533–536.

[341] T. Konstantin Rusch, Siddhartha Mishra, N. Benjamin Erichson, and Michael W. Mahoney. "Long Expressive Memory for Sequence Modeling." In: *arXiv:2110.04744 [cs, math, stat]* (Feb. 2022). arXiv: 2110.04744. URL: http://arxiv.org/abs/2110.04744.

[342] Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. "How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 3118–3135. DOI: 10.18653/v1/2021.acl-long.243.

[343] O. E. Rössler. "An equation for continuous chaos." In: *Physics Letters A* 57.5 (July 12, 1976), pp. 397–398. ISSN: 0375-9601. DOI: 10.1016/0375-9601(76)90101-8.

[344] Alaa Sagheer, Hala Hamdoun, and Hassan Youness. "Deep LSTM-Based Transfer Learning Approach for Coherent Forecasts in Hierarchical Time Series." eng. In: *Sensors (Basel, Switzerland)* 21.13 (June 2021), p. 4379. ISSN: 1424-8220. DOI: 10.3390/s21134379.

[345] David Salinas, Valentin Flunkert, and Jan Gasthaus. *DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks*. Feb. 22, 2019. DOI: 10.48550/arXiv.1704.04110. arXiv: 1704.04110[cs,stat].

[346] Lena Sasal, Tanujit Chakraborty, and Abdenour Hadid. *W-Transformers : A Wavelet-based Transformer Framework for Univariate Time Series Forecasting*. Sept. 8, 2022. URL: http://arxiv.org/abs/2209.03945.

[347] Tim Sauer. "Reconstruction of dynamical systems from interspike intervals." In: *Physical Review Letters* 72.24 (1994), p. 3811.

[348] Tim Sauer. "Interspike interval embedding of chaotic signals." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 5.1 (1995), pp. 127–132.

[349] Tim Sauer, James A Yorke, and Martin Casdagli. "Embedology." In: *Journal of statistical Physics* 65.3 (1991), pp. 579–616.

[350] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks." In: *Proceedings of the 2nd International Conference on Learning Representations*. 2014. URL: http://arxiv.org/abs/1312.6120.

[351] Raphael Sayer. "Bayesian Variational Inference for Piecewise-Linear Recurrent Neural Networks." Master's Thesis. Heidelberg, Germany: University of Heidelberg, 2020.

[352] Gerwin Schalk, Dennis J. McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R. Wolpaw. "BCI2000: a general-purpose brain-computer interface (BCI) system." In: *IEEE transactions on bio-medical engineering* 51.6 (June 2004), pp. 1034–1043. ISSN: 0018-9294. DOI: 10.1109/TBME.2004.827072.

[353] Steven J. Schiff, Kristin Jerger, Duc H. Duong, Taeun Chang, Mark L. Spano, and William L. Ditto. "Controlling chaos in the brain." en. In: *Nature* 370.6491 (Aug. 1994). Number: 6491 Publisher: Nature Publishing Group, pp. 615–620. ISSN: 1476-4687. DOI: 10.1038/370615a0.

[354] Jackie Schiller, Guy Major, Helmut J. Koester, and Yitzhak Schiller. "NMDA spikes in basal dendrites of cortical pyramidal neurons." en. In: *Nature* 404.6775 (2000), pp. 285–289. ISSN: 1476-4687. DOI: 10.1038/35005094.

[355] Dominik Schmidt, Georgia Koppe, Zahra Monfared, Max Beutelspacher, and Daniel Durstewitz. "Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies." In: *Proceedings of the 9th International Conference on Learning Representations*. 2021.

[356] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. "Learnable latent embeddings for joint behavioural and neural analysis." en. In: *Nature* 617.7960 (May 2023). Number: 7960 Publisher: Nature Publishing Group, pp. 360–368. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06031-6.

[357] Alexander Schuckert, Annabelle Bohrdt, Eleanor Crane, and Michael Knap. "Probing finite-temperature observables in quantum simulators of spin systems with short-time dynamics." In: *Physical Review B* 107.14 (Apr. 2023). Publisher: American Physical Society, p. L140410. DOI: 10.1103/PhysRevB.107.L140410.

[358] Alexander Schuckert, Izabella Lovas, and Michael Knap. "Nonlocal emergent hydrodynamics in a long-range quantum spin system." In: *Physical Review B* 101.2 (Jan. 2020). Publisher: American Physical Society, p. 020416. DOI: 10.1103/PhysRevB.101.020416.

[359] Terrence J. Sejnowski. *The Deep Learning Revolution*. en. The MIT Press, Oct. 2018. ISBN: 978-0-262-34682-5. DOI: 10.7551/mitpress/11474.001.0001.

[360] Terrence J. Sejnowski. "The unreasonable effectiveness of deep learning in artificial intelligence." en. In: *Proceedings of the National Academy of Sciences* 117.48 (Dec. 2020), pp. 30033–30038. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1907373117.

[361] Jussi Seppälä et al. "Mobile Phone and Wearable Sensor-Based mHealth Approaches for Psychiatric Disorders and Symptoms: Systematic Review." eng. In: *JMIR mental health* 6.2 (Feb. 2019), e9819. ISSN: 2368-7959. DOI: 10.2196/mental.9819.

[362] Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. *Bounding and Counting Linear Regions of Deep Neural Networks*. en. Sept. 2018. URL: http://arxiv.org/abs/1711.02114.

[363] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. "Learnability, Stability and Uniform Convergence." In: *Journal of Machine Learning Research* 11.90 (2010), pp. 2635–2670. ISSN: 1533-7928. URL: http://jmlr.org/papers/v11/shalev-shwartz10a.html.

[364] Srinivas Gorur Shandilya and Marc Timme. "Inferring network topology from complex dynamics." en. In: *New Journal of Physics* 13.1 (Jan. 2011), p. 013004. ISSN: 1367-2630. DOI: 10.1088/1367-2630/13/1/013004.

[365] Yuge Shi, Brooks Paige, Philip H. S. Torr, and N. Siddharth. "Relating by Contrasting: A Data-efficient Framework for Multimodal Generative Models." In: *arXiv:2007.01179 [cs, stat]* (Apr. 21, 2021). arXiv: 2007.01179. URL: http://arxiv.org/abs/2007.01179.

[366] Yuge Shi, N. Siddharth, Brooks Paige, and Philip H. S. Torr. "Variational mixture-of-experts autoencoders for multi-modal deep generative models." In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 15718–15729.

[367] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer." In: *SIAM Journal on Computing* 26.5 (Oct. 1997). Publisher: Society for Industrial and Applied Mathematics, pp. 1484–1509. ISSN: 0097-5397. DOI: 10.1137/S0097539795293172.

[368] Brian M. de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Nathan Kutz, and Steven L. Brunton. "PySINDy: A Python package for the Sparse Identification of Nonlinear Dynamics from Data." en. In: *arXiv preprint arXiv:2004.08424* (2020). URL: http://arxiv.org/abs/2004.08424.

[369] Adam Smith, M. S. Kim, Frank Pollmann, and Johannes Knolle. "Simulating quantum many-body dynamics on a current digital quantum computer." en. In: *npj Quantum Information* 5.1 (Nov. 2019). Publisher: Nature Publishing Group, pp. 1–13. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0217-0.

[370] Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. *Simplified State Space Layers for Sequence Modeling*. arXiv:2208.04933 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2208.04933.

[371] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. Nov. 2015. DOI: 10.48550/arXiv.1503.03585.

[372] Justin Solomon. *PDE Approaches to Graph Analysis*. arXiv:1505.00185 [cs, math]. Apr. 2015. DOI: 10.48550/arXiv.1505.00185.

[373] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. *Score-Based Generative Modeling through Stochastic Differential Equations*. arXiv:2011.13456 [cs, stat]. Feb. 2021. DOI: 10.48550/arXiv.2011.13456.

[374] Michael J. Sorocky, Siqi Zhou, and Angela P. Schoellig. "Experience Selection Using Dynamics Similarity for Efficient Multi-Source Transfer Learning Between Robots." In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2020, pp. 2739–2745. DOI: 10.1109/ICRA40945.2020.9196744.

[375] Sigurd Spieckermann, Siegmund Düll, Steffen Udluft, Alexander Hentschel, and Thomas Runkler. "Exploiting similarity in system identification tasks with recurrent neural networks." In: *Neurocomputing*. Learning for Visual Semantic Understanding in Big Data 169 (Dec. 2015), pp. 343–349. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2014.11.074.

[376] Kartik Sreenivasan, Jy-yong Sohn, Liu Yang, Matthew Grinde, Alliot Nagle, Hongyi Wang, Eric Xing, Kangwook Lee, and Dimitris Papailiopoulos. "Rare Gems: Finding Lottery Tickets at Initialization." en. In: May 2022. URL: https://openreview.net/forum?id=Jpxd93u2vK-.

[377] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. ISSN: 1533-7928. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[378] Martin Stemmler and Christof Koch. "How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate." en. In: *Nature Neuroscience* 2.6 (1999), pp. 521–527. ISSN: 1546-1726. DOI: 10.1038/9173.

[379] Sandra Steyaert, Marija Pizurica, Divya Nagaraj, Priya Khandelwal, Tina Hernandez-Boussard, Andrew J Gentles, and Olivier Gevaert. "Multimodal data fusion for cancer biomarker discovery with deep learning." In: *Nature Machine Intelligence* 5.4 (2023), pp. 351–362.

[380] M. Storace and O. De Feo. "Piecewise-linear approximation of nonlinear dynamical systems." In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.4 (Apr. 2004), pp. 830–842. ISSN: 1558-0806. DOI: 10.1109/TCSI.2004.823664.

[381] Steven Strogatz. *Infinite Powers: How Calculus Reveals the Secrets of the Universe*. en. Houghton Mifflin Harcourt, 2019. ISBN: 978-1-328-87998-1.

[382] Abhinav Subramanian and Sankaran Mahadevan. "Probabilistic physics-informed machine learning for dynamic systems." en. In: *Reliability Engineering & System Safety* 230 (Feb. 2023), p. 108899. ISSN: 0951-8320. DOI: 10.1016/j.ress.2022.108899.

[383] Mustafa Suleyman. *The Coming Wave: Technology, Power, and the Twenty-first Century's Greatest Dilemma*. en. Crown, Sept. 2023. ISBN: 978-0-593-59396-7.

[384] Christopher Summerfield and Kevin Miller. "Computational and systems neuroscience: The next 20 years." In: *PLOS Biology* 21.9 (Sept. 2023), e3002306. ISSN: 1544-9173. DOI: 10.1371/journal.pbio.3002306.

[385] David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. "LFADS - Latent Factor Analysis via Dynamical Systems." In: *arXiv:1608.06315 [cs, q-bio, stat]* (Aug. 2016). URL: http://arxiv.org/abs/1608.06315.

[386] Ilya Sutskever, James Martens, and Geoffrey Hinton. "Generating text with recurrent neural networks." In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. Madison, WI, USA: Omnipress, June 2011, pp. 1017–1024. ISBN: 978-1-4503-0619-5.

[387] Thomas M. Sutter, Imant Daunhawer, and Julia E. Vogt. "Generalized Multimodal ELBO." In: *arXiv:2105.02470 [cs, stat]* (June 25, 2021). arXiv: 2105.02470. URL: http://arxiv.org/abs/2105.02470.

[388] Floris Takens. "Detecting strange attractors in turbulence." en. In: *Dynamical Systems and Turbulence, Warwick 1980*. Vol. 898. Springer, 1981, pp. 366–381. ISBN: 978-3-540-11171-9 978-3-540-38945-3.

[389] Sachin S. Talathi and Aniket Vartak. "Improving performance of recurrent neural network with relu nonlinearity." In: *Proceedings of the 4th International Conference on Learning Representations*. 2016. URL: http://arxiv.org/abs/1511.03771.

[390] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. *Long Range Arena: A Benchmark for Efficient Transformers*. arXiv:2011.04006 [cs]. Nov. 2020. DOI: 10.48550/arXiv.2011.04006.

[391] Graham W. Taylor and Geoffrey E. Hinton. "Factored conditional restricted Boltzmann Machines for modeling motion style." en. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal Quebec Canada: ACM, June 2009, pp. 1025–1032. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553505.

[392] Gerald Teschl. *Ordinary Differential Equations and Dynamical Systems*. en. American Mathematical Soc., Aug. 2012. ISBN: 978-0-8218-8328-0.

[393] Saurabh Singh Thakur and Ram Babu Roy. "Predicting mental health using smart-phone usage and sensor data." en. In: *Journal of Ambient Intelligence and Humanized Computing* 12.10 (Oct. 2021), pp. 9145–9161. ISSN: 1868-5145. DOI: 10.1007/s12652-020-02616-5.

[394] Janine Thome, Mathieu Pinger, Daniel Durstewitz, Wolfgang H. Sommer, Peter Kirsch, and Georgia Koppe. "Model-based experimental manipulation of probabilistic behavior in interpretable behavioral latent variable models." English. In: *Frontiers in Neuroscience* 16 (Jan. 2023). Publisher: Frontiers. ISSN: 1662-453X. DOI: 10.3389/fnins.2022.1077735.

[395] Janine Thome, Robert Steinbach, Julian Grosskreutz, Daniel Durstewitz, and Georgia Koppe. "Classification of amyotrophic lateral sclerosis by brain volume, connectivity, and network dynamics." en. In: *Human Brain Mapping* 43.2 (2022), pp. 681–699. ISSN: 1097-0193. DOI: 10.1002/hbm.25679.

[396] Wesley K. Thompson, Anda Gershon, Ruth O'Hara, Rebecca A. Bernert, and Colin A. Depp. "The prediction of study-emergent suicidal ideation in bipolar disorder: a pilot study using ecological momentary assessment data." eng. In: *Bipolar Disorders* 16.7 (Nov. 2014), pp. 669–677. ISSN: 1399-5618. DOI: 10.1111/bdi.12218.

[397] Chunwei Tian, Yong Xu, Wangmeng Zuo, Chia-Wen Lin, and David Zhang. "Asymmetric CNN for Image Superresolution." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.6 (June 2022), pp. 3718–3730. ISSN: 2168-2232. DOI: 10.1109/TSMC.2021.3069265.

[398] Adam Tsakalidis et al. "Overview of the CLPsych 2022 Shared Task: Capturing Moments of Change in Longitudinal User Posts." In: *Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology*. Ed. by Ayah Zirikly et al. Seattle, USA: Association for Computational Linguistics, July 2022, pp. 184–198. DOI: 10.18653/v1/2022.clpsych-1.16.

[399] Belinda Tzen and Maxim Raginsky. *Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit*. Oct. 2019. DOI: 10.48550/arXiv.1905.09883.

[400] Eli Tziperman, Harvey Scher, Stephen E. Zebiak, and Mark A. Cane. "Controlling Spatiotemporal Chaos in a Realistic El Niño Prediction Model." In: *Phys. Rev. Lett.* 79 (6 1997), pp. 1034–1037. DOI: 10.1103/PhysRevLett.79.1034.

[401] Silviu-Marian Udrescu and Max Tegmark. "AI Feynman: A physics-inspired method for symbolic regression." In: *Science Advances* 6.16 (Apr. 2020). Publisher: American Association for the Advancement of Science. DOI: 10.1126/sciadv.aay2631.

[402] Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. *Deep learning generalizes because the parameter-function map is biased towards simple functions*. 2019. arXiv: 1805.08522 [stat.ML].

[403] Guillermo Valle-Pérez and Ard A. Louis. *Generalization bounds for deep learning*. arXiv:2012.04115 [cs, stat]. Dec. 2020. URL: http://arxiv.org/abs/2012.04115.

[404] Vincent Valton, Toby Wise, and Oliver J. Robinson. *Recommendations for Bayesian hierarchical model specifications for case-control studies in mental health*. arXiv:2011.01725 [cs, stat]. Nov. 2020. DOI: 10.48550/arXiv.2011.01725.

[405] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. New York, NY: Springer, 2000. ISBN: 978-1-4419-3160-3 978-1-4757-3264-1. DOI: 10.1007/978-1-4757-3264-1.

[406] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. Dec. 5, 2017. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762[cs].

[407] Sarah Jo C. Venditto, Kevin J. Miller, Carlos D. Brody, and Nathaniel D. Daw. "Dynamic reinforcement learning reveals time-dependent shifts in strategy during reward learning." eng. In: *bioRxiv* (Mar. 2024). DOI: 10.1101/2024.02.28.582617.

[408] Pantelis R. Vlachas, Wonmin Byeon, Zhong Y. Wan, Themistoklis P. Sapsis, and Petros Koumoutsakos. "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks." en. In: *Proc. R. Soc. A.* 474.2213 (2018), p. 20170844. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2017.0844.

[409] Pantelis R. Vlachas, Jaideep Pathak, Brian R. Hunt, Themistoklis P. Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. "Backpropagation Algorithms and Reservoir Computing in Recurrent Neural Networks for the Forecasting of Complex Spatiotemporal Dynamics." In: *arXiv:1910.05266 [physics]* (Feb. 2020). arXiv: 1910.05266. URL: http://arxiv.org/abs/1910.05266.

[410] 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). ISSN: 2297-4687. URL: https://www.frontiersin.org/articles/10.3389/fams.2022.818799.

[411] Henning U. Voss, Jens Timmer, and Jürgen Kurths. "Nonlinear dynamical system identification from uncertain and indirect measurements." In: *International Journal of Bifurcation and Chaos* 14.6 (June 2004), pp. 1905–1933. ISSN: 0218-1274. DOI: 10.1142/S0218127404010345.

[412] Grace Wahba. *Spline models for observational data*. SIAM, 1990.

[413] E.A. Wan and R. Van Der Merwe. "The unscented Kalman filter for nonlinear estimation." en. In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Lake Louise, Alta., Canada: IEEE, 2000, pp. 153–158. ISBN: 978-0-7803-5800-3. DOI: 10.1109/ASSPCC.2000.882463.

[414] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. "Generalizing to Unseen Domains: A Survey on Domain Generalization." In: *IEEE Transactions on Knowledge and Data Engineering* 35.8 (Aug. 2023), pp. 8052–8072. ISSN: 1558-2191. DOI: 10.1109/TKDE.2022.3178128.

[415] Rui Wang, Yihe Dong, Sercan Ö Arik, and Rose Yu. *Koopman Neural Forecaster for Time Series with Temporal Distribution Shifts*. en. Oct. 2022. URL: http://arxiv.org/abs/2210.03675.

[416] YingFei Wang, XiaoQun Wu, Hui Feng, JunAn Lu, and JinHu Lü. "Topology inference of uncertain complex dynamical networks and its applications in hidden nodes detection." en. In: *Science China Technological Sciences* 59.8 (Aug. 2016), pp. 1232–1243. ISSN: 1869-1900. DOI: 10.1007/s11431-016-6050-1.

[417] Elisa Warner, Joonsang Lee, William Hsu, Tanveer Syeda-Mahmood, Charles Kahn, Olivier Gevaert, and Arvind Rao. *Multimodal Machine Learning in Image-Based and Clinical Biomedicine: Survey and Prospects*. Publication Title: arXiv e-prints ADS Bibcode: 2023arXiv231102332W. Nov. 2023. DOI: 10.48550/arXiv.2311.02332.

[418] Duncan J. Watts and Steven H. Strogatz. "Collective dynamics of 'small-world' networks." In: *Nature* 393.6684 (1998), pp. 440–442. DOI: 10.1038/30918.

[419] Cheng K Fred Wen, Stefan Schneider, Arthur A Stone, and Donna Spruijt-Metz. "Compliance With Mobile Ecological Momentary Assessment Protocols in Children and Adolescents: A Systematic Review and Meta-Analysis." In: *Journal of Medical Internet Research* 19.4 (Apr. 2017), e132. ISSN: 1439-4456. DOI: 10.2196/jmir.6641.

[420] 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.

[421] Ronald J Williams and David Zipser. "A learning algorithm for continually running fully recurrent neural networks." In: *Neural computation* 1.2 (1989), pp. 270–280.

[422] H. R. Wilson and J. D. Cowan. "Excitatory and inhibitory interactions in localized populations of model neurons." In: *Biophysical Journal* 12.1 (Jan. 1972), pp. 1–24. ISSN: 0006-3495. DOI: 10.1016/S0006-3495(72)86068-5.

[423] Christopher Winship and Robert D. Mare. "Regression Models with Ordinal Variables." In: *American Sociological Review* 49.4 (1984). Publisher: [American Sociological Association, Sage Publications, Inc.], pp. 512–525. ISSN: 0003-1224. DOI: 10.2307/2095465.

[424] Alan Wolf, Jack B. Swift, Harry L. Swinney, and John A. Vastano. "Determining Lyapunov exponents from a time series." In: *Physica D: Nonlinear Phenomena* 16.3 (July 1, 1985), pp. 285–317. ISSN: 0167-2789. DOI: 10.1016/0167-2789(85)90011-9.

[425] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. Jan. 7, 2022. URL: http://arxiv.org/abs/2106.13008.

[426] Mike Wu and Noah Goodman. "Multimodal Generative Models for Scalable Weakly-Supervised Learning." In: *arXiv:1802.05335 [cs, stat]* (Nov. 12, 2018). arXiv: 1802.05335. URL: http://arxiv.org/abs/1802.05335.

[427] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. *Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case*. Jan. 22, 2020. arXiv: 2001.08317[cs,stat]. URL: http://arxiv.org/abs/2001.08317.

[428] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. "A Comprehensive Survey on Graph Neural Networks." In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (Jan. 2021), pp. 4–24. ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2020.2978386.

[429] Filip Wudarski, Daniel O'Connor, Shaun Geaney, Ata Akbari Asanjan, Max Wilson, Elena Strbac, P. Aaron Lott, and Davide Venturelli. *Hybrid quantum-classical reservoir computing for simulating chaotic systems*. en. arXiv:2311.14105 [quant-ph]. Nov. 2023. URL: http://arxiv.org/abs/2311.14105.

[430] Meiling Xu, Min Han, C. L. Philip Chen, and Tie Qiu. "Recurrent Broad Learning Systems for Time Series Prediction." In: *IEEE Transactions on Cybernetics* 50.4 (Apr. 2020), pp. 1405–1417. ISSN: 2168-2275. DOI: 10.1109/TCYB.2018.2863020.

[431] Peng Xu, Xiatian Zhu, and David A. Clifton. "Multimodal Learning With Transformers: A Survey." English. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.10 (Oct. 2023), pp. 12113–12132. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2023.3275156.

[432] Chenguang Yang, Chuize Chen, Wei He, Rongxin Cui, and Zhijun Li. "Robot Learning System Based on Adaptive Neural Control and Dynamic Movement Primitives." In: *IEEE Transactions on Neural Networks and Learning Systems* 30.3 (Mar. 2019), pp. 777–787. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2018.2852711.

[433] Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. "ADAHESSIAN: An Adaptive Second Order Optimizer for Machine Learning." en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.12 (May 2021). Number: 12, pp. 10665–10673. ISSN: 2374-3468. DOI: 10.1609/aaai.v35i12.17275.

[434] Chin-Chia Michael Yeh et al. *Toward a Foundation Model for Time Series Data*. arXiv:2310.03916 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2310.03916.

[435] Jun Yin and Yan Meng. "Self-organizing reservior computing with dynamically regulated cortical neural networks." In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407. June 2012, pp. 1–7. DOI: 10.1109/IJCNN.2012.6252772.

[436] Yuan Yin, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Baskiotis, and Patrick Gallinari. *LEADS: Learning Dynamical Systems that Generalize Across Environments*. 2022. arXiv: 2106.04546 [cs.LG].

[437] Shigeo Yoden. "Classification of simple low-order models in geophysical fluid dynamics and climate dynamics." In: *Nonlinear Analysis: Theory, Methods & Applications* 30.7 (1997). Proceedings of the Second World Congress of Nonlinear Analysts, pp. 4607–4618. ISSN: 0362-546X. DOI: https://doi.org/10.1016/S0362-546X(97)00306-4.

[438] Çağatay Yıldız, Markus Heinonen, and Harri Lähdesmäki. *ODE2VAE: Deep generative second order ODEs with Bayesian neural networks*. arXiv:1905.10994 [cs, stat]. Oct. 2019. DOI: 10.48550/arXiv.1905.10994.

[439] Peter Zeidman, Amirhossein Jafarian, Mohamed L. Seghier, Vladimir Litvak, Hayriye Cagnan, Cathy J. Price, and Karl J. Friston. "A guide to group effective connectivity analysis, part 2: Second level analysis with PEB." eng. In: *NeuroImage* 200 (Oct. 2019), pp. 12–25. ISSN: 1095-9572. DOI: 10.1016/j.neuroimage.2019.06.032.

[440] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. *Are Transformers Effective for Time Series Forecasting?* Aug. 2022. DOI: 10.48550/arXiv.2205.13504.

[441] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. *Understanding deep learning requires rethinking generalization*. en. arXiv:1611.03530 [cs]. Feb. 2017. URL: http://arxiv.org/abs/1611.03530.

[442] Jiayang Zhang, Jie Zhou, Ming Tang, Heng Guo, Michael Small, and Yong Zou. "Constructing ordinal partition transition networks from multivariate time series." In: *Scientific reports* 7.1 (2017), p. 7795.