Università degli Studi di Padova

Universitat Pompeu Fabra Barcelona

University of Padua
Department of Physics and Astronomy "G. Galilei"
Master Thesis in Physics of Data

Latent space modelling of whole-brain dynamics: a Koopman-theoretical approach

Degree candidate

Riccardo Tancredi

ID 2089395

Supervisor

Dr. Michele Allegra

University of Padua

Co-supervisor

Prof. Gustavo Deco

Univesitat Pompeu Fabra, Barcelona

Academic Year 2024/2025

Contents

Abstract **v**

1 Data-driven embeddings **3**
1.1 The Koopman paradigm 4
1.1.a Koopman operator theory 4
Spectral decomposition 6
Full spectrum 7
Koopman eigenfunction group 8
Principal eigen-pairs 8
Example 8
1.1.b Discover Koopman operators 10
1.2 Neural network approaches 13
1.2.a Neural Networks 13
1.2.b Auto-encoders 13
1.3 Koopman operator estimates 16
1.3.a Unconstrained Koopman operator 18
1.3.b Constrained Koopman operator 19
Eigenvalues parametrization 19
1.3.c Eigenfunctions approach 20
1.4 Kernel-based Koopman operator 20

2 Real Dataset **23**
2.1 Imaging Methods 25
2.1.a Functional Magnetic Resonance 25
2.1.b Applications of fMRI 25
2.1.c MRI Procedure and Scanning 26
2.2 Data selection and filtering 27

3 Analysis of synthetic data **29**
3.1 Harmonic Oscillators 30
3.1.a $d = 1$ 30
3.1.b $d \gg 1$ 32
3.2 Simple model with discrete spectrum 33
3.3 Hopf bifurcation model 36

4 Analysis of real data **41**
4.1 Latent dimension estimation 41
4.2 Per-subject analysis 43
4.3 All-subjects analysis 47
4.3.a Homotopic connectivity 51
4.4 Principal Component Analysis (PCA) 53
4.4.a PCA results 54
4.5 Conclusions 57
4.5.a Synthetic Data Analysis 57
4.5.b Analysis of Real Data 58
4.5.c Model Selection and Latent Space Considerations 58
4.5.d Implications and Future Directions 59

Bibliography **61**

Abstract

Koopman theory offers a powerful framework for transitioning from complex, finite-dimensional and non-linear representation of physical systems to infinite-dimensional but linear ones. Over the last few decades, this theory has gained renewed attention and interest with the possibility to linearize physical systems by identifying key modes of the Koopman operator by training Deep Neural Networks. In this work, Koopman linear-based embeddings are developed through auto-encoders to achieve coordinate transformations and dimensionality reductions that enable linear approximations of non-linear dynamics. Several Neural Network architectures have been tested on both discrete and continuous spectrum problems, proving the versatility of this paradigm. Finally, a real-world fMRI dataset of stroke patients is analysed to evaluate the model's robustness to noise and variability, as well as its ability to predict future system states.

Introduction

A dynamical system is traditionally described in terms of its state space, where at any given time t_0 the state of the system is represented as a point in a geometrical manifold. The evolution of the system is typically modelled by a function that maps the current state at t_0 to future states for $t > t_0$. Several challenges can arise when studying such systems:

- the possible absence of a complete analytical description;
- the presence of non-linearities, which generally hinder an exhaustive characterisation of the system's trajectories;
- the high dimensionality of the state space, which complicates its representation and interpretation;
- inherent complexities, such as rapidly varying dynamics and unforeseen behaviour, that further complicate the analysis.

In this work, we address these challenges by adopting a data-driven approach to reduce the high-dimensional description of a system to a lower-dimensional latent manifold. In this latent space, a few key coordinates are identified which, ideally, allow the system's evolution to be described by linear dynamics. A central tool in this endeavour is the Koopman operator [1], which facilitates the representation of non-linear dynamics through an infinite-dimensional linear framework. Recent advances in machine learning have enabled the estimation of Koopman eigenfunctions and eigenvalues directly from data, thereby opening up new avenues for prediction, control, and analysis of complex systems.

The thesis is organised into four main chapters:

1. **Data-driven embeddings**: this chapter introduces the theoretical foundations of data-driven embedding techniques, including auto-encoder architectures and the application of Koopman operator theory to obtain linear representations of non-linear systems.

2. **Real dataset**: in this chapter, we present the real-world fMRI dataset used in this study, detailing its acquisition, preprocessing, and a preliminary analysis of its properties.

3. **Analysis of synthetic data**: here, we apply the proposed methods to synthetic data in order to validate the framework under controlled conditions and benchmark its performance against known analytical solutions.

4. **Analysis on real data**: finally, we apply the developed techniques to the real fMRI dataset, investigating the latent representations, dynamic modes, and their implications for understanding brain dynamics, particularly in the context of stroke.

Throughout these chapters, we aim to demonstrate that a carefully designed data-driven embedding approach can yield a low-dimensional representation in which the complex, non-linear dynamics of the original system are approximately linearised. This, in turn, facilitates effective analysis, prediction, and potentially control of the system.

1 Data-driven embeddings

In this chapter, we present a comprehensive data-driven framework for embedding high-dimensional non-linear dynamical systems into a lower-dimensional latent space where the dynamics are rendered approximately linear. Our approach is rooted in Koopman operator theory, originally introduced by Koopman [1] in his seminal work on transformations of Hamiltonian systems in Hilbert space, which established the possibility of representing a finite non-linear system by an infinite-dimensional linear operator.

To render the Koopman operator practically useful, subsequent research [2, 3] has focused on obtaining finite-dimensional approximations by leveraging its eigenfunctions. More recently, advances in machine learning have enabled the estimation of these eigenfunctions via neural networks, thereby facilitating applications in prediction, linear control, and system analysis [4, 5]. Furthermore, techniques such as Dynamic Mode Decomposition (DMD) [6, 7] offer an alternative, data-driven means to extract coherent spatio-temporal patterns directly from time-series data, complementing Koopman analysis.

In the following sections, we introduce the fundamental principles underlying Koopman operator theory – including its linearity, infinitesimal generator, spectral decomposition, and the construction of Koopman eigenfunctions – as well as the role of basis functions in forming finite-dimensional representations of infinite-dimensional operators. We also discuss how these concepts integrate with modern deep learning approaches, such as auto-encoder-based embeddings, to provide a unified framework for analysing complex dynamical systems.

Overall, this chapter lays the theoretical and methodological groundwork for our data-driven embedding approach, demonstrating its potential for reconstructing the intrinsic dynamics of the system and enabling efficient prediction

and control strategies.

1.1 The Koopman paradigm

Koopman operator theory was first introduced by B.O. Koopman in a paradigmatic work [1] on transformations of Hamiltonian systems in Hilbert space. The overarching idea there introduced was the possibility to represent a finite non-linear system by an infinite-dimensional linear one. The existence of such an operator had a crucial role in proving the Mean Ergodic Theorem of von Neumann [8] and Birkhoff [9]. The difficulty of estimating an infinite dimensional operator in Hilbert space led Mézic et al. [2, 3] to focus on the research of finite-dimensional representations of the Koopman operator using its eigenfunctions. In recent years the techniques developed for machine learning have led to the possibility of estimating such eigenfunctions using neural networks or similar processes for prediction, linear control and analysis [4]. The spectrum of the operator allows decomposing non-linear systems into different dynamic regimes, while linearly evolving coordinates contain intrinsic resourceful information and properties of the system relevant for the analysis. Moreover, the linear behaviour of the new coordinates alleviates the task of model predictive control through numerical integration [10].

These approaches have open up new perspectives, but all the proposed Koopman operator realizations are still profoundly data-driven and a general paradigm for estimating its eigenfunctions and eigenvalues is missing. The non-triviality of this process has led to various Koopman operator constructions, simplifying different properties of the operator itself and consequently its eigenfunctions.

1.1.a Koopman operator theory

Consider a continuous time dynamical system

$$\dot{\boldsymbol{x}} = \boldsymbol{\phi}(\boldsymbol{x}), \tag{1.1}$$

and a scalar observable function $\psi : \mathcal{V} \to \mathbb{C}$

$$y = \psi(\boldsymbol{x}), \tag{1.2}$$

on a smooth M-dimensional manifold $\mathcal{V}$, $\boldsymbol{x} \in \mathcal{V}$, $t \in \mathbb{R}_{+,0}$ with a smooth and Lipschitz flow $\boldsymbol{\Phi}^t : \mathcal{V} \to \mathcal{V}$, such that $\boldsymbol{\Phi}^t(\boldsymbol{x})$ has a unique solution in $[0, +\infty)$

from a initial condition $\boldsymbol{x}_0$ at time t_0:

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad \boldsymbol{\Phi}^t := \boldsymbol{x}_0 + \int_{t_0}^{t_0+t} \boldsymbol{\phi}(\boldsymbol{x}(\tau))\, d\tau. \tag{1.3}$$

In this context, an observable can be any kind of a measurement of the system.

The semigroup of Koopman operators $\{\mathcal{K}^t\}_{t\in\mathbb{R}_{+,0}} : \mathcal{F} \to \mathcal{F}$ acts on scalar observables functions $\psi : \mathcal{V} \to \mathbb{C}$ by composition with the flow semigroup $\{\boldsymbol{\Phi}^t\}_{t\in\mathbb{R}_{0,+}}$ of the vector field $\boldsymbol{\phi}$

$$\mathcal{K}^t_{\boldsymbol{\phi}}\psi = \psi \circ \boldsymbol{\Phi}^t, \tag{1.4}$$

on the state space $\mathcal{V}$, where "$\circ$" represents function composition. When dealing with a discrete-time system, such as those occurring when a non-analytical and numerical treatment is adopted, one can write

$$\mathcal{K}_{\boldsymbol{\Phi}}\psi = \psi \circ \boldsymbol{\Phi}, \tag{1.5}$$

dropping the superscript t in $\boldsymbol{\Phi}^t$. Several significant properties follow from this definition.

Linearity

The Koopman operator is, by construction, a linear operator. Consider the Koopman operator $\mathcal{K}_{\boldsymbol{\Phi}}$ and two observables $g_1, g_2 \in \mathcal{F}$ and two scalars $\alpha, \beta \in \mathbb{R}$. Then, it follows that

$$\begin{aligned} \mathcal{K}_{\boldsymbol{\Phi}}(\alpha g_1 + \beta g_2) &= (\alpha g_1 + \beta g_2) \circ \boldsymbol{\Phi} \\ &= \alpha g_1 \circ \boldsymbol{\Phi} + \beta g_2 \circ \boldsymbol{\Phi} \\ &= \alpha \mathcal{K}_{\boldsymbol{\Phi}} g_1 + \beta \mathcal{K}_{\boldsymbol{\Phi}} g_2, \end{aligned} \tag{1.6}$$

which is the definition of a linear operator.

Infinitesimal generator

There exists a generator $\mathcal{G}_{\mathcal{K}} : D \to \mathcal{F},\ D \subset \mathcal{F}$ being the domain of the generator and $\mathcal{F}$ the Banach space of the observables.

The operator $\mathcal{G}_{\mathcal{K}}$ is the infinitesimal generator of the time-t indexed semigroup of Koopman operators $\{\mathcal{K}^t\}_{t\in\mathbb{R}_{+,0}}$

$$\mathcal{G}_{\mathcal{K}}\psi = \lim_{t\to 0^+} \frac{\mathcal{K}^t\psi - \psi}{t} = \frac{d}{dt}\psi. \tag{1.7}$$

For a bounded generator $\mathcal{G}_\mathcal{K}$ one is able to get the time-t semigroup of Koopman operators as $\mathcal{K}^t = \exp(\mathcal{G}_\mathcal{K} t)$. Similarly to the Koopman operator, the same approach can be adopted to compute the linear evolution of the observables for the Koopman operator generator.

Koopman eigenfunctions

An observable $\varphi \in \mathcal{F}$ is an eigenfunction if it satisfies:

$$\begin{aligned} \mathcal{G}_\mathcal{K}\varphi(\boldsymbol{x}) &= \dot{\varphi}(\boldsymbol{x}) \\ &= s\varphi(\boldsymbol{x}), \end{aligned} \tag{1.8}$$

$s \in \mathbb{C}$ being the associated eigenvalue.

Linear coordinates

Since $\mathcal{G}_\mathcal{K}$ is the infinitesimal generator of the Koopman operators' semigroup $\{\mathcal{K}^t\}_{t\in\mathbb{R}_{+,0}}$, the following is also satisfied for $\lambda^t = \exp(st)$:

$$\mathcal{K}^t_{\boldsymbol{\phi}}\varphi(\boldsymbol{x}) = \varphi(\boldsymbol{\Phi}^t(\boldsymbol{x})) = \lambda^t\varphi(\boldsymbol{x}), \tag{1.9}$$

along the vector field's flow. For the traditional discrete-time case, the Koopman operator $\mathcal{K}_\boldsymbol{\Phi}$ composes an observable with the discrete and time non-parametrized map $\boldsymbol{x} \mapsto \boldsymbol{\Phi}(\boldsymbol{x})$: in this case it is possible to retrieve a more common eigenvalue equation:

$$\mathcal{K}_\boldsymbol{\Phi}\varphi(\boldsymbol{x}) = \varphi(\boldsymbol{\Phi}(\boldsymbol{x})) = \lambda\varphi(\boldsymbol{x}), \tag{1.10}$$

with $\lambda = \exp(st_s)$, where t_s is the sampling time of the discrete-time dynamical system.

It is immediate to understand that within this paradigm, any arbitrary observable can be time-evolved if we are able to express it as a linear combination of Koopman eigenfunctions.

Spectral decomposition

An arbitrary observable can be written via the eigenfunctions of the Koopman operator:

$$\psi(\boldsymbol{x}) = \sum_{j=1}^{\infty} v_j(\psi)\varphi_j(\boldsymbol{x}) \tag{1.11}$$

where v_j are coefficients resembling modes of the corresponding eigenfunctions φ_j of the operator $\mathcal{K}$.

It is important to define Koopman modes with respect to eigenfunctions, rather than eigenvalues (which can be degenerate). For instance, let the eigenvalue λ have a 2-dimensional eigenspace $\mathcal{E}_\lambda$ with basis $\{\varphi_1, \varphi_2\}$, and let us consider vector-valued observables. If we define $\psi = (f_1,\ f_2)^\top$ with $f_1 = c_1\varphi_1 + c_2\varphi_2$ and $f_2 = c_3\varphi_2$ scalar-value functions, the Koopman modes associated to φ_1 and φ_2 are:

$$v_1(\psi) = \begin{pmatrix} c_1 \\ 0 \end{pmatrix}, \qquad v_2(\psi) = \begin{pmatrix} c_2 \\ c_3 \end{pmatrix} \tag{1.12}$$

respectively.

The infinite sum appearing in Eq. (1.11) is, of course, related to the infinite-dimensional nature of the Koopman operator. This spectral decomposition, which expresses any observable as a linear combination of Koopman eigenfunctions, not only captures the global linear dynamics but also preserves the invariance of the coordinates in the operator space. The action of the Koopman operator on an observable such as $\psi(\boldsymbol{x})$ is better explicated by the following equation:

$$\mathcal{K}^t\psi = \sum_{j=1}^{\infty} v_j(\psi)\left[\mathcal{K}^t\varphi_j\right] = \sum_{j=1}^{\infty} v_j(\psi)\lambda_j^t\varphi_j \tag{1.13}$$

where $\lambda_j^t = \exp(s_j t)$, with $s_j = \mu_j + i\omega_j$ being a complex number whose real part, μ_j, represents the eigen-decay or eigen-growth rate along the j-th component, while its imaginary part, ω_j, corresponds to the eigen-frequency of oscillations in that subspace.

Full spectrum

In general, the spectral decomposition can account for a continuous spectrum as well [3]. Assuming that the system has an attractor $\mathcal{A}$ with a preserved measure $d(\mathcal{A})$, we can write the $\mathcal{K}$ operator decomposition as

$$\mathcal{K}^t\psi(\boldsymbol{x}) = \underbrace{\sum_{j=1}^{\infty} v_j(\psi)\lambda_j^t\varphi_j(\boldsymbol{x})}_{\text{discrete spectrum}} + \underbrace{\int_0^{2\pi} \kappa^t E(d\kappa)\psi(\boldsymbol{x})}_{\text{continuous spectrum}} \tag{1.14}$$

where $\kappa^t = \exp(i\theta t)$ is the integrated function over frequency θ and $E(\kappa)$ the projection-valued measure.

While from the discrete spectrum it is feasible to extract both transient and oscillatory components of the dynamics, $\mathcal{R}e(\lambda_j^t)$ and $\mathcal{I}m(\lambda_j^t)$ respectively, the

continuous part encodes the more chaotic component of the dynamics, corresponding to the on-attractor dynamics.

Koopman eigenfunction group

Under the assumption that the space $\mathcal{F}$ is chosen to be a Banach algebra, e.g. $C^1(\mathcal{V})$, the set of eigenfunctions forms an *Abelian* semigroup under the point-wise product of functions [4]. Products of eigenfunctions are themselves eigenfunctions. Therefore, it holds that if $\varphi_1, \varphi_2 \in \mathcal{F}$ are eigenfunctions of the composition operator $\mathcal{K}_{\mathbf{\Phi}}$ with eigenvalues λ_1, λ_2, then the product $\varphi_1\varphi_2$ is still an eigenfunction of $\mathcal{K}_{\mathbf{\Phi}}$ with eigenvalue $\lambda_1\lambda_2$.

Principal eigen-pairs

Consider $\{E^m\}_{m\in\mathbb{N}}$ to be the eigen-pair semigroup of Koopman operator $\mathcal{K}_{\mathbf{\Phi}}$. $\{E^m\}$ has a minimal generator $\mathcal{G}_E$, such that:

$$\{E^m\} = \left\{ \left(\prod_{j=1}^{m} \lambda_j^{n_j}, \prod_{j=1}^{m} \varphi_j^{n_j} \right) \middle| (\lambda_j, \varphi_j) \subset \mathcal{G}_E \right\}, \tag{1.15}$$

where $m, n_j \in \mathbb{N}$. The elements of $\mathcal{G}_E$ are called *principal eigen-pairs*, and they form the minimal set of Koopman operator eigen-pairs that can be used to construct all other eigen-pairs [4].

Example

For the sake of clarity, we revisit a well-known example from the literature, considering a simple non-linear system with a single fixed point and a discrete eigenvalue spectrum.
Let the map $\mathbf{\Phi} : \mathbb{R}^2 \to \mathbb{R}^2$:

$$\boldsymbol{x} \mapsto \mathbf{\Phi}(\boldsymbol{x}) = \begin{pmatrix} ax_1 \\ bx_2 + \left(b - a^2\right)x_1^2 \end{pmatrix} \tag{1.16}$$

where $a, b \in [0,\, 1]$. Imposing $\mathbf{\Phi}(\boldsymbol{x}) = 0$, one can easily find that the system has a stable equilibrium at the origin and an invariant manifold given by $(x_1,\, x_2)^\top = (x_1,\, -x_1^2)^\top$.

It can be computed that the *principal* eigenvalue-eigenfunction pairs $(\boldsymbol{\lambda}, \boldsymbol{\varphi}(\boldsymbol{x}))$ of the system are (a, x_1) and $(b, x_2 + x_1^2)$ and using the *Abelian* semigroup property under point-wise products of functions, it is possible to obtain more eigenfunction-eigenvalue pairs as powers of the *principal* ones, e.g. (a^2, x_1^2) as

the product of (a, x_1) with itself. In this manner, the system can be lifted into a new, linear one with coordinates:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \begin{pmatrix} x_1 \\ x_2 + x_1^2 \\ x_1^2 \end{pmatrix} \mapsto \underbrace{\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & a^2 \end{pmatrix}}_{\boldsymbol{\Lambda}} \boldsymbol{\varphi}(\boldsymbol{x}). \tag{1.17}$$

and it is immediate to notice that $\boldsymbol{\varphi}(\boldsymbol{x})$ is a vector of eigenfunctions of $\boldsymbol{\Lambda}$.

This result can then be projected onto the vector of *output functions*, e.g. $\psi(\boldsymbol{x}) = (x_1,\ x_2)^\top$ via modes of $\mathcal{K}$ operator:

$$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \tag{1.18}$$

such that

$$\mathcal{K}_{\boldsymbol{\Phi}}(\boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{\varphi}(\boldsymbol{x}) \tag{1.19}$$

with $\boldsymbol{V} = (v_1,\ v_2,\ v_3)$, offering an equivalent representation of the original 2-dimensional system. At this point, the system's propagation does not necessitate the composition of non-linear maps but only an initial non-linear transformation $\boldsymbol{\varphi}$ after which the evolution is linear:

$$\boldsymbol{\Phi}^{(k)}(\boldsymbol{x}) = \boldsymbol{V}\boldsymbol{\Lambda}^{(k)}\boldsymbol{\varphi}(\boldsymbol{x}) \tag{1.20}$$

where k-compositions of non-linear functions are replaced with the $\boldsymbol{\varphi}$-embedding and subsequent linear evolution through the composition operator paradigm.

Of course, the coordinates proposed above are not necessarily the optimal ones, as achieving a linear description of the dynamics does not require finding coordinates that are exact eigenfunctions of the operator, but rather ones that lie within the span of the eigenfunctions.

Indeed, for the same system, the choice

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \begin{pmatrix} x_1 \\ x_2 \\ x_1^2 \end{pmatrix} \mapsto \underbrace{\begin{pmatrix} a & 0 & 0 \\ 0 & b & (b - a^2) \\ 0 & 0 & a^2 \end{pmatrix}}_{\boldsymbol{A}} \boldsymbol{\varphi}(\boldsymbol{x}). \tag{1.21}$$

still leads to a linear mapping, even though not all coordinates are eigenfunctions, i.e. x_2. This result can then be projected onto the system states using the output

matrix $\boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$:

$$\mathcal{K}_{\boldsymbol{\Phi}}(\boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{A}\boldsymbol{\varphi}(\boldsymbol{x}). \tag{1.22}$$

Here the columns of $\boldsymbol{C}$ do not represent Koopman modes, but by diagonalizing $\mathcal{K}$ into $\boldsymbol{\Lambda}$, one uncovers Koopman operator eigenvalues, eigenfunctions and modes. For linear prediction, it is immediate to use

$$\boldsymbol{\Phi}^{(k)}(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{A}^{(k)}\boldsymbol{\varphi}(\boldsymbol{x}) \tag{1.23}$$

which is equivalent to the formulation presented before.

1.1.b Discover Koopman operators

Recalling the definition of the Koopman operator, $\mathcal{K}$ is generally infinite-dimensional, as it operates on a function space. The inability to handle an infinite-dimensional representation of the operator forces us to consider different approaches to identify a finite description of the operator.

Several methods have been proposed in recent years to obtain finite-dimensional representations of the Koopman operator [6, 10, 11]. A common approach involves projecting the operator onto a predefined function basis. However, the arbitrariness of this choice often requires an infinite number of basis functions to ensure closure within that space. Instead, selecting a basis that is intrinsically closed under the Koopman operator allows for a well-defined finite-dimensional representation, making a Koopman-invariant basis feasible. Nevertheless, a set of Koopman-invariant coordinates is generally not sufficient for a proper reconstruction of the system's dynamics, which requires an infinite number of coordinates. Learning adequate coordinates via *lifting* functions that linearise the dynamics might be a strategy, but is a non-trivial approach.

Dynamic Mode Decomposition

Dynamic Mode Decomposition (DMD) is a data-driven technique for analysing the spatio-temporal evolution of dynamical systems, introduced in fluid dynamics and later extended to various domains such as neuroscience and control theory. The core idea behind DMD is to approximate the underlying dynamics of a system through a linear operator that best captures the evolution of snapshots of data over time [6, 12].

Consider a sequence of measurements (or snapshots) of a system at discrete

time steps:

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m \in \mathbb{R}^n. \tag{1.24}$$

These measurements are typically collected into two data matrices:

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_{m-1} \end{bmatrix}, \quad \mathbf{X}' = \begin{bmatrix} \boldsymbol{x}_2 & \boldsymbol{x}_3 & \ldots & \boldsymbol{x}_m \end{bmatrix}. \tag{1.25}$$

DMD seeks a best-fit linear operator $\mathbf{A}$ such that:

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X}. \tag{1.26}$$

This operator is generally not directly accessible but can be approximated via the least-squares solution:

$$\mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger}, \tag{1.27}$$

where $\mathbf{X}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{X}$.

To extract the DMD modes and their associated dynamical behaviour, one typically performs an eigendecomposition of $\mathbf{A}$. Since $\mathbf{A}$ is often high-dimensional, a reduced representation is obtained via Singular Value Decomposition (SVD). Specifically, if:

$$\mathbf{X} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\top}, \tag{1.28}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\boldsymbol{\Sigma}$ contains the singular values, then $\mathbf{A}$ is projected onto the lower-dimensional space as:

$$\tilde{\mathbf{A}} = \mathbf{U}^{\top}\mathbf{A}\mathbf{U}. \tag{1.29}$$

The eigenvalues λ_j of $\tilde{\mathbf{A}}$ correspond to the DMD eigenvalues, which describe the temporal evolution of the modes. Each eigenvalue is associated with a DMD mode $\boldsymbol{\phi}_j$, which provides spatial information:

$$\mathbf{A}\boldsymbol{\phi}_j = \lambda_j \boldsymbol{\phi}_j. \tag{1.30}$$

DMD has been extended in various ways to accommodate control inputs, time-varying systems, and non-linear phenomena. One such extension, DMD with control (DMDc), incorporates exogenous inputs into the dynamics, modifying Eq. (1.26) to:

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}, \tag{1.31}$$

where $\mathbf{U}$ represents control inputs, and $\mathbf{B}$ is an additional operator learned from data [7].

Overall, DMD provides an interpretable, data-driven framework for studying complex dynamical systems by decomposing them into spatio-temporal modes governed by a low-rank linear approximation. It serves as a fundamental tool for bridging machine learning techniques and dynamical systems theory [12].

High dimensional embedding

Data-driven *embeddings* of the data might allow capturing the non-linearity in the dynamics. The idea is to develop a way to *lift* data through a map $\boldsymbol{\varphi}$. $\mathcal{K}$ can then be estimated by solving the following optimization problem, inspired by the approach used in DMD methods:

$$\min_{\mathcal{K},\boldsymbol{C},\boldsymbol{\varphi}} \sum_{i=1}^{T-1} \underbrace{\left\|\boldsymbol{z}_{i+1} - \mathcal{K}\boldsymbol{z}_i\right\|^2}_{\text{linearity}} + \underbrace{\left\|\boldsymbol{y}_i - \boldsymbol{C}\boldsymbol{z}_i\right\|^2}_{\text{reconstruction}}, \tag{1.32}$$

$$\boldsymbol{y}_i = \boldsymbol{\Phi}(\boldsymbol{x}_i), \tag{1.33}$$

$$\boldsymbol{z}_i = \boldsymbol{\varphi}(\boldsymbol{x}_i), \tag{1.34}$$

$$\boldsymbol{\varphi} \in \mathcal{F}, \tag{1.35}$$

$\mathcal{F}$ being the space of observables and T the length of the observed trajectory. If no *embedding* is used, i.e. $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x}$, this is of course equivalent to traditional DMD. The target representation is in the form of

$$\boldsymbol{z}_0 = \boldsymbol{\varphi}(\boldsymbol{x}_0), \tag{1.36}$$

$$\boldsymbol{z}_{t+1} = \mathcal{K}z_t, \tag{1.37}$$

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{z}_t, \tag{1.38}$$

where the initial lifting (1.36) leads to linear state-transition coordinates (1.37), then projected on the output of interest (1.38).

Some extended dynamic mode decomposition (EDMD) fix a lifting map, i.e. Eq. (1.34), and then aim to solve Eq. (1.32) as a convex, least-squares optimization problem of $\mathcal{K}$ and $\boldsymbol{C}$.

1.2 Neural network approaches

The previously discussed DMD models often suffer from inefficiency, as they provide only locally accurate predictions. To overcome this limitation, techniques such as those proposed by Li et al. [13] and Lusch et al. [5] propose learning the mapping in Eq. (1.34) simultaneously with the optimization of the operator, formulating a problem closely related to Eq. (1.32).

In this work, we will delve into the possibilities presented in [5], where a sophisticated neural network is trained to address the aforementioned problems, and the map defined in Eq. (1.38) is not necessarily linear, allowing for the representation of a wider range of dynamical regimes. This specific modification may actually simplify the problem under study, as it can lead to low-dimensional embeddings, exploiting auto-encoders.

1.2.a Neural Networks

In the last fifteen years, deep learning [14] techniques have broadened the horizon of research, enabling to learn approximations of unknown functions or system descriptors directly from data.

Neural networks (*NNs*), which form the theoretical architecture of deep learning, were originally inspired by the primary visual cortex of cats where neurons are organized in hierarchical layers of cells to process visual stimuli [15]. The first mathematical model of a *NN* was the neocognitron [16] which has many of the features of modern deep neural networks (*DNNs*), such as multi-layer structure, convolution, non-linear dynamical nodes. Above all, in the mathematical framework of artificial neural networks, universal approximation theorems [17, 18] state that a family of neural networks, sufficiently wide and sufficiently deep, approximate any function arbitrarily well. Formally, for each function f from a certain function space, there exists a sequence of neural networks $NN_1, NN_2, \ldots,$ from the family, such that $NN_n \to f$ according to some criterion. This endows neural networks with the potential to approximate any type of dynamical system, allowing the extraction of relevant features directly from data.

1.2.b Auto-encoders

In many real-world scenarios, not only do we lack an analytical description of the system, but the system itself is also high-dimensional and observed through complex, non-linear transformations. Consider a measurement apparatus that records a set of M observable features that are related to physical variables of interest –such as temperature, atmospheric pressure, and UV index – only

through complex non-linear relations. In this case, the relevant variables would not be directly represented in the recorded state, but only latent, and difficult to isolate.

This issue is particularly relevant when dealing with high-dimensional systems, where the observed variables may appear complex, even though an underlying simpler structure might be revealed through appropriate embeddings. In such cases, extracting meaningful, low-dimensional representations that retain the essential structure of the system is crucial. Auto-encoders provide a powerful framework for learning a reduced-order latent space representation, encoding an M-dimensional state into an N-dimensional latent space, where $N < M$, while preserving the most informative features relevant for downstream analysis.

An auto-encoder [19] is an artificial neural network that learns two functions:

- an encoding function $\boldsymbol{\varphi} : \mathcal{M} \to \mathcal{N}$ that maps the M-dimensional input data onto a latent N-dimensional manifold;
- a decoding function $\boldsymbol{\psi} : \mathcal{N} \to \mathcal{M}$ that recreates the M-dimensional input data from the N-dimensional latent representation.

Called $\boldsymbol{x} \in \mathcal{M} \subset \mathbb{R}^M$ the input real vector, the encoder produces a vector

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{y} \in \mathcal{N} \subset \mathbb{R}^N, \tag{1.39}$$

while the decoder

$$\boldsymbol{x}' = \boldsymbol{\psi}(\boldsymbol{y}) \in \mathcal{M} \subset \mathbb{R}^M \tag{1.40}$$

outputs a vector in the high-dimensional vector space.

In principle, since we are only encoding and decoding the data, we expect

$$\boldsymbol{x}' \equiv \boldsymbol{x} \tag{1.41}$$

and this is achieved if

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{\psi}(\boldsymbol{y}) \\ &= \boldsymbol{\psi}(\boldsymbol{\varphi}(\boldsymbol{x})) \qquad \iff \boldsymbol{\psi} \equiv \boldsymbol{\varphi}^{-1} \\ &\overset{!}{=} \boldsymbol{x}' \end{aligned} \tag{1.42}$$

therefore solving the following minimization problem:

$$\min_{\boldsymbol{\varphi}} \sum_{i=1}^{T} \left\| \boldsymbol{x}_i - \boldsymbol{\varphi}^{-1}(\boldsymbol{\varphi}(\boldsymbol{x}_i)) \right\|^2, \tag{1.43}$$

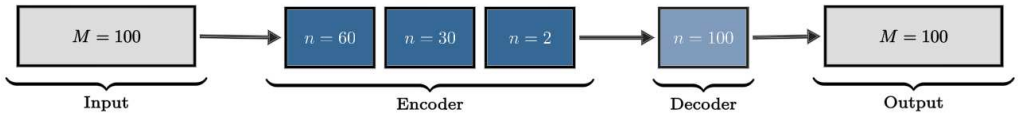

Figure 1.1: Auto-encoder architecture: the input and output have a dimension of $M = 100$. The encoder consists of three layers with widths $n = 60,\ 30,\ 2$, respectively. The decoder is a simple linear layer that reconstructs the signal from the latent space back to the original space.

where T is the dimension of the input data. The functions $\boldsymbol{\varphi}$ are constrained by the architecture of the neural network. In particular, they belong to a specific function space determined by the network's structure and activation functions. If the network consists only of linear transformations (i.e. no non-linear activation functions), then $\boldsymbol{\varphi}$ reduce to a linear map, which in the case $N < M$ is a projection. Otherwise, the choice of non-linear activation functions determines a restricted function space from which $\boldsymbol{\varphi}$ are drawn, preventing the trivial identity solution in Eq. (1.43).

Example

We present a simple example to demonstrate how auto-encoders are capable of embedding data and compressing the information of a system into meaningful coordinates.

To illustrate this concept in a simple yet rigorous manner, consider a one-dimensional harmonic oscillator:

$$\ddot{x} = -\omega^2 x \implies \begin{cases} \dot{x} = y \\ \dot{y} = -\omega^2 x \end{cases} \tag{1.44}$$

embedded in an M-dimensional space ($M \gg 2$). This high-dimensional embedding is achieved by multiplying a matrix $W \in \mathcal{M}_{M\times 2}$ by the vector $(x,\ y)^\top$.

We want the auto-encoder to find both the encoding and decoding functions, mapping the M-dimensional input signal into a latent space of dimension $N = 2$ of coordinates $\boldsymbol{\varphi}(\boldsymbol{x}) = (\tilde{x},\ \tilde{y})^\top$. In this example, M is set to 100 and $\omega = 2\, rad/s$.

The chosen architecture for the neural network is depicted in Fig. (1.1), featuring an encoder having a funnel-shaped structure, consisting of three hidden layers with widths 60, 30 and 2, respectively, and utilizing a `ReLU` activation function to achieve a non-linear mapping into the latent manifold. The decoder, on the other hand, is a single linear layer of dimension M that, as in Eq. (1.38),

maps the problem back in a linear fashion, mimicking the action of the W matrix.

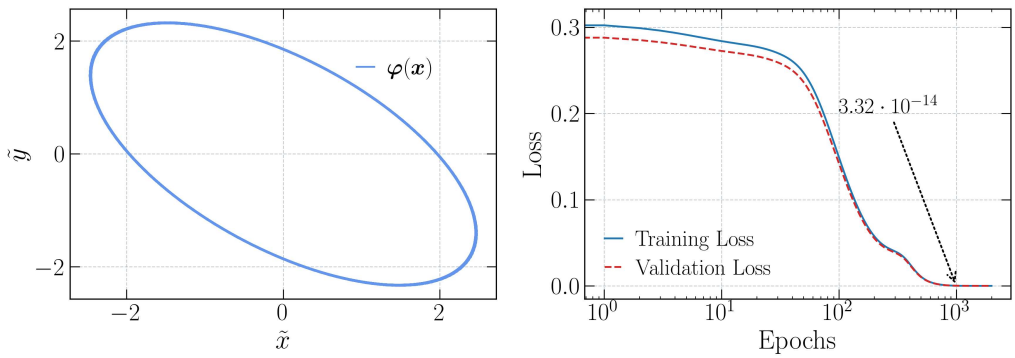

Figure 1.2: Auto-encoder performance: encoded coordinates $\varphi(\boldsymbol{x})$ on the embedded manifold (left). Mean squared error (loss) for both training and validation sets as a function of epochs (right).

As shown, both the training and test losses are very small, indicating that the reconstruction is highly accurate. Furthermore, the plot demonstrates that the encoder has successfully reproduced the two coordinates x and $\dot{x}$ of the harmonic oscillator, except for a linear transformation of the phase space.

1.3 Koopman operator estimates

To better capture the system's dynamics in the latent space, we extend the auto-encoder architecture by introducing an additional layer, positioned immediately after the encoder. This layer is designed to estimate the Koopman operator, represented by the matrix $\mathcal{K} \in \mathcal{M}_{N\times N}$, which governs the time evolution of the latent representations. By enforcing a linear transition in this space, we aim to extract meaningful spectral properties that characterize the system's behaviour over time.

Following the schema in Fig. (1.3), once the encoder and decoder depths and widths have been decided, the Koopman layer can be as simple as a linear layer of dimension $N \times N$ (N being the dimension of the latent space). Therefore, $\mathcal{K}$ is a matrix representing the linear evolution of the system:

$$\varphi(\boldsymbol{x}_{t+1}) = \mathcal{K}\varphi(\boldsymbol{x}_t), \quad t = 1, \ldots, T-1, \tag{1.45}$$

where T is the number of time steps in the considered trajectory and $\varphi(\boldsymbol{x})$ is the encoder function, satisfying Eq. (1.36).

To properly estimate $\mathcal{K}$ as well as both the encoder φ and the decoder φ^{-1} functions, we modified the optimization problem outlined in Eq. (1.32). Specifically,

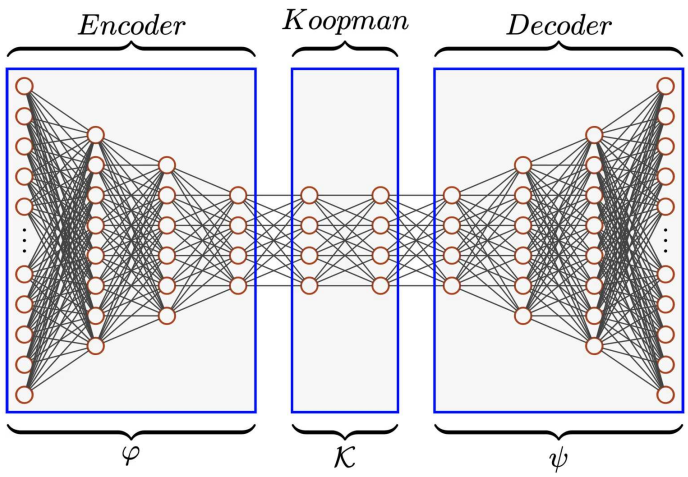

Figure 1.3: Neural network scheme: the encoder layers, denoted as $\boldsymbol{\varphi}$, transform the input signal into the latent space. The resulting representation is then processed by the Koopman layers, $\mathcal{K}$, before being mapped back to the original space through the decoder layers, $\boldsymbol{\psi} \equiv \boldsymbol{\varphi}^{-1}$.

we account for the possible issue of continuous eigenvalues, i.e. cases where eigenvalues vary along the trajectory. As a result, $\mathcal{K}_t$ is treated as a function of the embedded coordinates $\boldsymbol{\varphi}(\boldsymbol{x}_t)$.

The neural network loss function is therefore composed of three terms:

1. **Reconstruction** term: we seek to identify the intrinsic coordinates $\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{x})$ where the dynamics evolve, in combination with the inverse mapping $\boldsymbol{x} = \boldsymbol{\varphi}^{-1}(\boldsymbol{y})$ to retrieve the original state $\boldsymbol{x}$. This is achieved using the auto-encoder loss, i.e. Eq. (1.43):

$$\mathcal{L}_{recon} = \frac{1}{T}\sum_{t=1}^{T}\left\|\boldsymbol{x}_t - \boldsymbol{\varphi}^{-1}(\boldsymbol{\varphi}(\boldsymbol{x}_t))\right\|_{MSE}, \tag{1.46}$$

   where *MSE* refers to the mean square error. The dimension N of the latent space is a hyperparameter of the network, which should be fixed, based on any prior knowledge of the system.

2. **Linear** term: to discover the Koopman operator and its eigenvalues, we learn the linear dynamics in the latent space via Eq. (1.45). This is achieved using the following *MSE* loss:

$$\left\|\boldsymbol{\varphi}(\boldsymbol{x}_{t+1}) - \mathcal{K}_t\boldsymbol{\varphi}(\boldsymbol{x}_t)\right\|_{MSE}, \ \forall t = 1, \dots, T-1.$$

To enforce linear prediction in the intrinsic space, we extend the prediction over m time steps:

$$\mathcal{L}_{lin} = \frac{1}{T-1}\sum_{t=2}^{T}\left\|\boldsymbol{\varphi}(\boldsymbol{x}_t) - \prod_{m=1}^{t-1}\left[\mathcal{K}_{t-m}\right]\boldsymbol{\varphi}(\boldsymbol{x}_1)\right\|_{MSE}, \tag{1.47}$$

where $\prod_{m=1}^{t-1}\left[\mathcal{K}_{t-m}\right]$ represents m-times the action of the Koopman operator on the coordinate $\boldsymbol{\varphi}(\boldsymbol{x}_1)$. If the system's spectrum is discrete, the loss term can be simplified considering that the Koopman operator is the same for each point:

$$\prod_{m=1}^{t-1}\left[\mathcal{K}_{t-m}\right] \equiv \mathcal{K}^{t-1} \tag{1.48}$$

3. **Prediction** term: we additionally impose that the latent coordinates allow us to predict the system's future states in the original space. This leads to the *MSE* loss

$$\left\|\boldsymbol{x}_{t+1} - \boldsymbol{\varphi}^{-1}(\mathcal{K}_t\boldsymbol{\varphi}(\boldsymbol{x}_t))\right\|_{MSE}, \ \forall t = 1, \ldots, T-1,$$

and considering, as before, the prediction over m consecutive time steps:

$$\mathcal{L}_{pred} = \frac{1}{S_p-1}\sum_{t=2}^{S_p}\left\|\boldsymbol{x}_t - \boldsymbol{\varphi}^{-1}\left(\prod_{m=1}^{t-1}\left[\mathcal{K}_{t-m}\right]\boldsymbol{\varphi}(\boldsymbol{x}_1)\right)\right\|_{MSE} \tag{1.49}$$

The total loss $\mathcal{L}$, resulting from the combination of all the terms introduced

$$\mathcal{L} = \alpha_1\mathcal{L}_{recon} + \alpha_2\mathcal{L}_{lin} + \alpha_3\mathcal{L}_{pred}, \tag{1.50}$$

represents the optimization problem we aim to solve. Given this loss, several approaches to finding a suitable coordinate representation and estimate the Koopman operator are possible, depending on the constraints imposed on $\boldsymbol{\varphi}$ and $\mathcal{K}$. Here, we propose three paradigms based on auto-encoders with and without constraints on $\mathcal{K}$.

1.3.a Unconstrained Koopman operator

This is one of the most naive approaches, as the Koopman operator acts as a simple linear mapping from the intrinsic coordinates $\boldsymbol{\varphi}(\boldsymbol{x}_k)$ at time step $t = k$ to their evolved counterpart $\boldsymbol{\varphi}(\boldsymbol{x}_{k+1})$ at time $t' = k+1$. Despite the simplicity

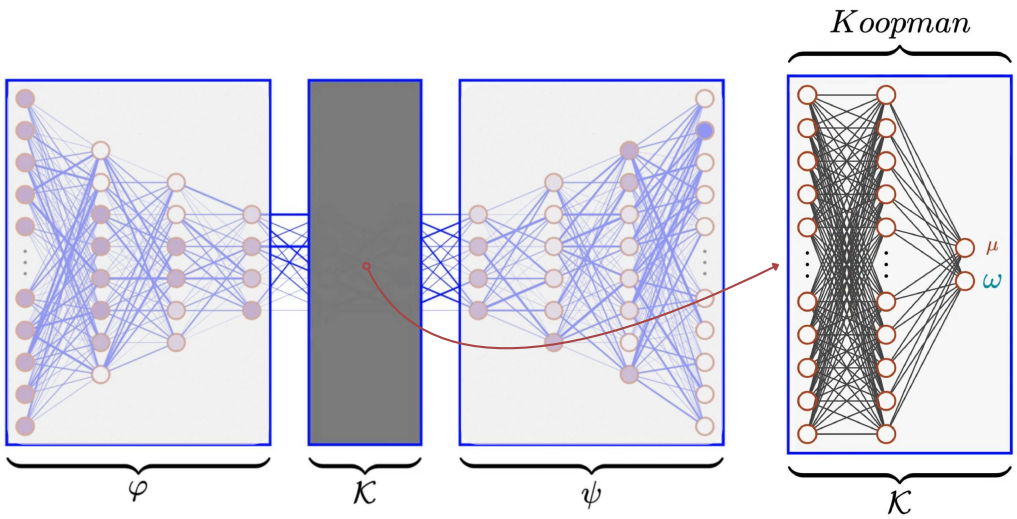

Figure 1.4: Auxiliary Network scheme: The Koopman layers are replaced by an auxiliary network that estimates the system eigenvalues $\boldsymbol{\mu}$ and $\boldsymbol{\omega}$ based on the embedded coordinates $\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{x})$. These eigenvalues form the diagonal blocks of the Koopman operator, capturing the underlying system dynamics.

of this approach, the eigenvalues of $\mathcal{K}$ can provide relevant information about the system if the linear approximation of the dynamics is valid. In the third chapter 3, we will illustrate the effectiveness of this approach by extending the discussion of the example introduced before (harmonic oscillator) and showing that this approach can retrieve the oscillatory frequencies of n different harmonic oscillators.

1.3.b Constrained Koopman operator

Eigenvalues parametrization

In their paper [5], Lusch, Kutz and Brunton introduced an innovative approach for estimating the Koopman operator $\mathcal{K}(\boldsymbol{\lambda})$ by parametrizing with continuously varying eigenvalues $\boldsymbol{\lambda}$. This is achieved by implementing a neural network that aims to identify the Koopman eigenfunctions $\boldsymbol{y} := \boldsymbol{\varphi}(\boldsymbol{x})$ in a latent space of fixed dimension, followed by an auxiliary network that estimates a set of characteristic frequencies, which are used to construct the Koopman operator. This approach allows mapping non-linear oscillators with continuous spectra into single pairs of conjugate eigenfunctions, representing trajectories as combinations of sines and cosines.

The eigenvalues of the matrix $\mathcal{K}$ are therefore allowed to vary, parametrized

by the function $\boldsymbol{\lambda} = \boldsymbol{\Lambda}(\boldsymbol{y})$. The eigenvalues $\boldsymbol{\lambda}(\boldsymbol{y})_{\pm} = \boldsymbol{\mu}(\boldsymbol{y}) \pm i\boldsymbol{\omega}(\boldsymbol{y})$ are then used to parametrize the Koopman matrix built as a block-diagonal matrix $\mathcal{K}(\boldsymbol{\mu}, \boldsymbol{\omega})$. Each pair of complex eigenvalues is represented in a Jordan block of the form:

$$B(\mu, \omega) = e^{(\mu \Delta t)} \begin{pmatrix} \cos(\omega \Delta t) & -\sin(\omega \Delta t) \\ \sin(\omega \Delta t) & \cos(\omega \Delta t) \end{pmatrix}, \tag{1.51}$$

Δt being the integration time. The Koopman operator is therefore a diagonal-block matrix where each block is of the form of Eq. (1.51) and such that $\mathcal{K} \in \mathcal{M}_{N \times N}$.

It is evident that the imaginary part ω of each eigenvalue λ represents a rotation in the phase space of the latent coordinates, while the real part μ indicates contraction or expansion, depending on its sign, in the embedded space.

1.3.c Eigenfunctions approach

Li and Lan [20] proposed a revised method that focuses primarily on the extraction of eigenfunctions of the Koopman operator rather than concentrating on its invariant subspace. The latter approach imposes a consistent number of constraints on the weight and size of the auxiliary network associated to the auto-encoder, leading to increased uncertainty and instability during training, especially when the dimension of the invariant space N is high. In contrast, Li and Lan focused on eigenfunctions, which are extracted directly by the auto-encoder in a step-by-step manner. The most important eigenfunctions – i.e. those that provide a comprehensive yet minimal representation of the system's dynamics, enabling an effective linearization of the original non-linear system – are the first ones to be obtained. This approach allows for a reduction in the size of the neural network and the number of parameters, effectively minimizing the risk of overfitting or getting trapped in local minima. However, the construction of the Koopman operator is very similar to the one proposed in [5].

1.4 Kernel-based Koopman operator

Basis functions learning

In [21], the authors proposed an experimental, data-driven signal processing method to compute the Koopman operator for the eigendecomposition of electrophysiological signals into orthogonal, coherent components and to examine their associated spatio-temporal dynamics. Their approach focused on extracting spatio-temporal patterns of neuronal dynamics through the operator-theoretic

formulation of ergodic theory [22].

To solve the Koopman eigenvalue problem stated in Eq. (1.45), they first constructed a data-driven basis of the Hilbert space $\mathcal{H}$ using a class of kernel-based algorithms, known as non-linear Laplacian spectral analysis (NLSA), which operates on delayed-embedding coordinates [23]. Instead of employing an auto-encoder framework, they defined an augmented observation map $F_Q : \mathcal{M} \to \mathbb{R}^{Qd}$, where Q represents the number of chosen delays and d the number of sensors on the cerebral cortex. A kernel function $k_Q : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$, such as a Gaussian kernel, is then introduced to measure pairwise similarity between points on the manifold $\mathcal{M}$. This kernel is used to construct a Markov kernel $p_{Q,T-Q}(\boldsymbol{x}_t, \boldsymbol{x}_{t'})$, which plays a crucial role in forming a $(T-Q) \times (T-Q)$ ergodic Markov matrix $P_{Q,T-Q}$. Here, $\boldsymbol{x} \in \mathcal{M}$ represents the dataset under consideration, and T denotes the number of snapshots in the dataset.

It can be shown that the eigen-pairs (λ_j, ϕ_j) solving the eigenvalue problem

$$P_{Q,T-Q}\, \phi_j = \lambda_j\, \phi_j$$

can be mapped to the eigenfunctions of an approximate Koopman operator.

While all these approaches offer valuable insights, the present work primarily focused on the framework developed in [5], adopting a constrained Koopman operator perspective. This choice is motivated by the hypothesis that the system's dynamics can be effectively reduced to a small number of dominant non-linear oscillators. Under this assumption, the approach proposed by Lusch et al. [5] appears particularly suitable, as it provides a structured way to extract meaningful oscillatory modes while enforcing consistency with the Koopman framework.

2 Real Dataset

The interest in Koopman theory arises primarily from its potential to linearize complex and non-linear dynamics, enabling the prediction of future states of a system. A functioning Koopman model, in principle, should be capable of addressing these challenges effectively. However, the application to real-world data is far from straightforward, as such data are often characterized by noise and experimental artifacts.

Chapter 4, which contains the core of the results of this thesis, focuses on the analysis of a real dataset, on which the predictive capabilities of the Koopman framework are evaluated. The present chapter is instead devoted to a detailed description of the dataset used, namely, the Washington University Stroke Cohort dataset [24]. This dataset comprises neuroimaging data acquired using $3T$ MRI and clinical information from patients who experienced a first-time stroke. The study followed patients at three time points: $1-2$ weeks ($\mu_{event} = 13.4$ days, $\sigma_{event} = 4.8$ days), 3 months, and 1 year post-stroke onset. Additionally, a group of 27 age-matched control subjects was evaluated twice at an interval of three months. A total of $n = 96$ subjects were recruited between May 2008 and May 2013 from the stroke service at Barnes-Jewish Hospital (BJH), with support from the Washington University Cognitive Rehabilitation Research Group (CRRG). Patients were included based on the following criteria:

- clinical diagnosis of stroke at hospital discharge;
- persistent stroke symptoms at hospital discharge;
- awake, alert, and able to complete study tasks;
- age 18 or older.

However, if they met any of the following criteria they were excluded:

- history of prior stroke, based on clinical imaging;
- multi-focal stroke, based on clinical diagnosis or imaging;
- more than two lacunes visible on head CT scans performed before study enrolment;
- diagnosis of schizophrenia, bipolar disorder, major depression, or other severe psychiatric conditions;
- dementia (as documented in medical records, indicated by a Short Blessed Score ≥ 9, or a premorbid AD-8 score ≥ 2);
- neurological disorders such as epilepsy or Parkinson's disease;
- brain injury;
- end-stage renal disease, terminal cancer, class III or IV heart failure, or other conditions with a life expectancy of less than one year;
- pre-morbid functional disability (Modified Rankin Score ≥ 2);
- claustrophobia or presence of metal objects precluding the use of $3T$ MRI.

From the CRRG Registry, 6260 charts were screened. Of these, 5438 (87%) were excluded based on chart review alone: 1697 were discharged with a diagnosis other than stroke; of the remaining candidates, 1181 had a prior history of stroke; then 1188 had other neurological or psychiatric history listed as exclusion criteria; then 877 had other medical co-morbidities listed as exclusion criteria; then 387 had contraindications for MRI, and finally 108 showed multi-focal lesions or excessive lacunae on clinical head CT. The 822 candidates who passed the screening were all approached for enrolment. Of these 822, 172 were enrolled in the study. The other 650, were transferred to a facility outside the university hospital system, or remained medically unstable during the enrolment window (634), or were randomized to enrolment in a competing protocol (16).

Of the 172 patients enrolled, 40 were later excluded: 14 were unable to tolerate the scan; 2 had artifacts on the scan; 12 had tiny or questionable lesions; and 12 were found to have multi-focal lesions or excessive lacunae.

However, for our studies, we used a subset of $n = 96$ patients. This selection was made to replicate the sampling described by Favaretto et al. [25]. In that work, the authors selected participants who had complete data for all time points (2 weeks, 3 months, 1 year) and who met specific quality criteria, such

as a minimum number of valid frames, following standard protocols to ensure data quality.

The healthy control group, initially composed of 30 participants was matched with the study sample for age, gender, and years of education. They serve as a performance baseline in the neuropsychological and behavioural battery. They were enrolled using the same exclusion criteria as the stroke study group, from neighbours or spouses of stroke study participants. However, only 27 participants were ultimately included in the analysis, following again the procedure described in [25].

2.1 Imaging Methods

2.1.a Functional Magnetic Resonance

Functional Magnetic Resonance Imaging (fMRI) is a non-invasive neuroimaging technique widely used to study brain activity by measuring changes in blood flow and oxygenation levels. Its foundation lies in the Blood Oxygenation Level-Dependent (BOLD) contrast, introduced by Ogawa et al. in 1990 [26]. This technique capitalizes on the paramagnetic properties of deoxygenated haemoglobin, which create local magnetic field inhomogeneities detectable by MRI scanners.

The BOLD signal reflects a combination of neural, vascular, and metabolic processes. When neurons in a specific brain region are active, they consume oxygen, leading to a localized increase in blood flow that surpasses oxygen consumption. This results in a relative increase in oxygenated haemoglobin, reducing local magnetic susceptibility and increasing the MRI signal intensity [27].

A key strength of fMRI is its ability to provide whole-brain coverage with a spatial resolution of $2-3mm$ and a temporal resolution dictated by the repetition time (TR) of image acquisition, typically ranging from 1 to 3 seconds [28]. This temporal resolution limits the technique's ability to capture rapid neural events but is sufficient for studying slower processes, such as those giving rise to resting-state networks [29].

2.1.b Applications of fMRI

fMRI is extensively used in both task-based and resting-state paradigms. In task-based fMRI, participants perform specific activities, such as motor or cognitive tasks, while neural activation is mapped to corresponding regions. Resting-state fMRI, on the other hand, captures spontaneous fluctuations in the BOLD signal

while the participant is not engaged in any task, revealing functional connectivity patterns [30].

While fMRI is a powerful tool, it comes with several limitations. First, the BOLD signal is an indirect measure of neural activity, heavily influenced by vascular and metabolic factors. Additionally, its temporal resolution is constrained by the haemodynamic response function, which lags neural activity by several seconds. Moreover, motion artifacts, susceptibility to noise, and variability in vascular responses between individuals pose challenges for data interpretation [29]. Finally, the technique requires careful preprocessing, including motion correction, spatial normalization, and filtering, to mitigate confounding effects and enhance signal quality.

2.1.c MRI Procedure and Scanning

MRI scanning was conducted using a Siemens $3T$ Tim-Trio scanner at the School of Medicine, Washington University in St. Louis [31, 32]. The imaging protocol included structural, functional, and diffusion tensor imaging. The structural scans consisted of:

- a sagittal MP-RAGE $T1$-weighted scan with parameters: repetition time ($TR = 1950\,ms$), echo time ($TE = 2.26\,ms$), flip angle ($\phi = 9°$), voxels dimensions ($s = 1.0 \times 1.0 \times 1.0\,mm^3$), and slice thickness ($t = 1.00\,mm$);

- a transverse turbo spin-echo $T2$-weighted scan with parameters: $TR = 2500\,ms$, $TE = 435\,ms$, voxels size $s = 1.0 \times 1.0 \times 1.0\,mm^3$, and slice thickness $t = 1.00\,mm$;

- a sagittal FLAIR (fluid attenuated inversion recovery) scan with parameters: $TR = 7500\,ms$, $TE = 326\,ms$, voxels size $s = 1.5 \times 1.5 \times 1.5\,mm^3$, and slice thickness $t = 1.50\,mm$.

Therefore, resting-state functional MRI scans were taken from each subject. Participants were instructed to fixate on a small centrally located white fixation cross that was presented against a black background on a screen at the back of the magnet bore. Seven resting-state scans (128 volumes each) were obtained from each participant, for a total of approximately 30 minutes giving a total of 896 time points for each participant.

The preprocessing of resting-state fMRI included:

- regression of head motion, signal from ventricles and CSF, signal from white matter, global signal;

- temporal filtering retaining frequencies in $I_f = [0.008, 0.08]\ Hz$ band. This frequency band is a standard for rs-fMRI studies;
- frame censoring, $FD = 0.5\ mm$.

Finally, the resulting time series were projected on the cortical and subcortical surface of each subject divided into 235 ROIs (200 cortical plus 35 subcortical regions of interest)[1]. These areas are taken from the multi-resolution functional connectivity-based cortical parcellations developed by Schaefer and colleagues [33], including additional subcortical and cerebellar parcels from the Automated Anatomical Labelling (AAL) atlas [34] and a brainstem parcel from the Harvard–Oxford Subcortical atlas.

2.2 Data selection and filtering

The selected data underwent a second-order ($k = 2$) Butterworth bandpass filter, a key preprocessing step for analysing time series data such as fMRI signals. This filter provides a flat frequency response in the passband, minimizing signal distortion. It is well-suited for isolating neural oscillations within the typical frequency range of resting-state fMRI signals: $I_f = [0.008, 0.08]\ Hz$. With a repetition time $TR = 2\ s$, the Nyquist frequency represents half the sampling frequency of the signal, defining the upper limit for accurately represented frequencies without aliasing: $f_{\text{Nyquist}} = \frac{1}{2 \cdot TR} = 0.25\ Hz$. Consequently, the normalized passband frequencies are $\mathbf{Wn} = \frac{I_f}{f_{\text{Nyquist}}} = [0.032, 0.32]$.

A relevant issue in the dataset was the presence of *not-a-number* (NaN) values. If an entire ROI contained NaN values throughout the sampling period, they were replaced with Gaussian noise, where the noise standard deviation σ_{noise} matched that of the signals from the previous ROI. For each region of interest, signals were first de-trended and demeaned. Gaussian noise with standard deviation σ_{noise} was added in the case of NaNs. Subsequently, a bandpass digital filter was applied in both forward and backward directions to ensure zero phase distortion [35].

Finally, z-score normalization was performed temporally to center the signal and standardize its variance. Out of the original 235 ROIs, 234 were retained for analysis, as the last one subcortical region exhibited significant artifacts in its structural connectivity.

[1]The atlas and the functional labels for each ROI can be obtained together with the scripts used for this preprocessing on GitHub.

The final dataset comprises 96 stroke patients and 27 healthy controls, totalling $n = 123$ subjects. Each subject is represented by time series data consisting of 896 time points across 234 regions of interest.

This neuroimaging dataset provides valuable insight into the brain's dynamics post-stroke. These data are essential for testing the Koopman framework's ability to identify intrinsic coordinates, predict recovery trajectories, compare stroke patients with healthy subjects, and uncover latent dynamical patterns.

3 Analysis of synthetic data

Understanding and modelling complex dynamical systems require methods that can extract meaningful latent representations while preserving the underlying system's evolution. However, when dealing with real-world data, evaluating the accuracy of a model can be challenging due to measurement noise, incomplete information, and the lack of ground truth for the system's latent dynamics. For this reason, synthetic datasets play a crucial role in validating the effectiveness of data-driven approaches before their application to real-world problems.

In this chapter, we employ synthetic data to systematically assess the capabilities of the constrained Koopman framework, presented in section (1.3.b). By using well-defined dynamical systems, we can directly compare the learned latent representations and Koopman operators against known analytical solutions, thus providing a controlled environment for model validation. This allows us to investigate several key aspects of the approach, including:

- the expressiveness of the learned latent space, to understand how well the model captures the system's essential dynamics;
- the reconstruction accuracy, to explore if the latent representation preserves the original system's trajectory;
- the spectral analysis, to investigate if the estimated Koopman eigenvalues and modes are meaningful and consistent with theoretical predictions.

To address these points, we analyse a series of test cases that range from simple systems, where the Koopman operator has an analytical representation, to more complex non-linear dynamics that challenge the model's ability to approximate the system's evolution. Each section of this chapter is dedicated to a specific

synthetic dataset, detailing the system's properties, the training setup, and the evaluation of the results.

Through these experiments, we aim to provide a rigorous assessment of the framework's potential while also identifying its boundaries. The insights gained from this analysis will guide the application of the method to real-world data in the following chapter.

3.1 Harmonic Oscillators

Starting from Eq. (1.44), which describes a one-dimensional harmonic oscillator, we introduce the Koopman framework to assess whether the auto-encoder, combined with the constrained approach (section (1.3.b)), can achieve a global linearization of the system and recover its key properties, such as the eigenvalues of the original dynamics.

In the following sections, we analyse different scenarios in which a system of d harmonic oscillators is linearly embedded into a higher-dimensional space. This embedding is performed by applying a transformation matrix $W \in \mathcal{M}_{M\times 2d}$ to the original $2d$-dimensional state vector, $(\boldsymbol{x}, \boldsymbol{y})_t^\top \in \mathbb{R}^{2d}$, $t = 1, \ldots, T$, where T denotes the length of one trajectory. Thus, the input signal for each *NN* in this section is given by the matrix $\mathbb{X} \in \mathbb{R}^{M\times T}$. Each harmonic oscillator in the system is characterized by its own oscillation frequency ω_i, $i = 1, \ldots, d$, collectively forming the discrete spectrum of the system. The goal of our Koopman analysis is to accurately retrieve this spectrum, providing insight into the system's fundamental modes.

3.1.a $d = 1$

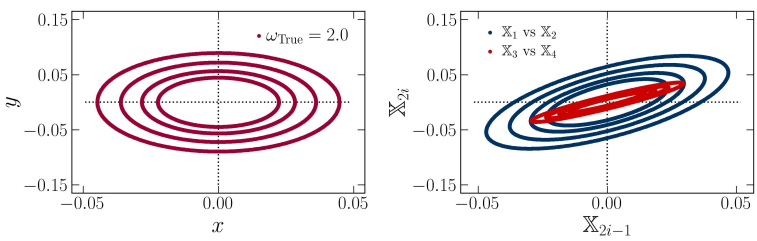

Figure 3.1: Original dataset $(\boldsymbol{x},\ \boldsymbol{y})^\top$(left) and two higher-order harmonic components $(i = 1, \ldots, \lfloor \frac{M}{2} \rfloor)$ obtained through the transformation $\mathbb{X} = W \cdot \boldsymbol{x}$ (right).

We first considered a single harmonic oscillator with a natural frequency of $\omega_{\text{True}} = 2.0\, rad/s$, on the left in Fig. (3.1), embedded into a higher-dimensional space of dimension $M = 100$ (right plot).

The training set consists of 300 trajectories, each starting from a different point in the phase space. The validation and test sets each contain 100 trajectories. The selected integration time step is $dt = 0.04$, resulting in trajectories of length $T = 101$ points. The hyperparameter in Eq. (1.49) is set to $S_p = 50$. The chosen encoder consists of 3 layers with dimensions $[32, 16, 2]$, each using a `ReLU` activation function, except for the last layer. The decoder is a mirrored version of the encoder architecture. The auxiliary network used to estimate the Koopman eigenvalues, as shown in Fig. (1.4), consists of a layer of size 200 with `ReLU` activation function, followed by an output layer of size 2 with a linear activation function.

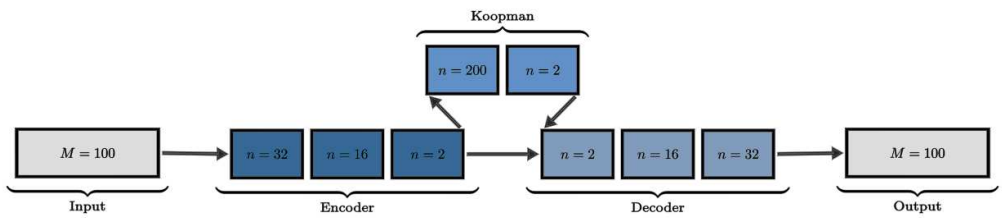

Figure 3.2: Neural Network architecture: the input and output dimensions are $M = 100$. The encoder consists of three layers with $n = 32, 16, 2$ neurons, respectively. The decoder is also a three-layer component, mirroring the encoder structure with $n = 2,\ 16,\ 32$ neurons per layer. Between the encoder and decoder, the Koopman auxiliary network is trained to estimate the system's eigenvalues. This network consists of two layers with $n = 200$ and $n = 2$ neurons, respectively. `ReLU` activation function is applied to all layers except the last layer of each component.

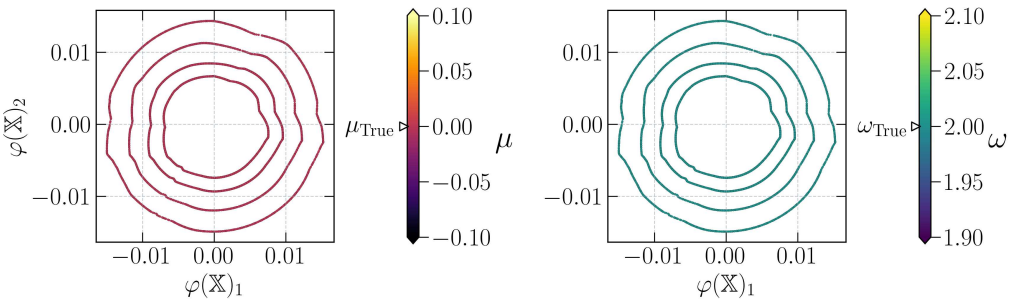

Figure 3.3: Linear embedding via $\mathcal{K}$ operator: embedded coordinates $\boldsymbol{\varphi}(\mathbb{X})$ of the harmonic oscillator. The plots show the estimated system eigenvalues, μ (left) and ω (right), for different trajectories in the latent space. The predictions closely match the parameters used in the simulation.

The latent space dimension is set to $N = 2$, as we expect to retrieve the eigenvalue pair $\lambda_{\pm} = \pm i\omega$. This example represents a combination of the strengths of auto-encoders and a first attempt at Koopman-based modelling. The goal

here is not a full global linearization but rather a simplified representation of the embedded coordinates $\mathbb{X}$, allowing for a linear integrator in terms of the two original coordinates $(\boldsymbol{x},\ \boldsymbol{y})^\top$. In a problem formulated as the one presented here, retrieving the oscillation frequency would be straightforward by performing a Fourier transform on the encoder output, as the one illustrated in Fig. (1.2). However, this approach does not allow for predicting the system's future states.

Fig. (3.3) shows the embedded coordinates $\boldsymbol{\varphi}(\mathbb{X})$. Using the method from [5], we retrieve the system's continuous eigenvalues, despite its discrete spectrum, as already noted. The method successfully isolates the system's only governing frequency ω, corresponding to an eigenvalue with zero real part μ.

The embedded coordinates do not form perfect ellipses, like those in the left plot of Fig. (3.1). This is likely an unavoidable consequence of the use of non-linear activation functions. In this specific example, given the linear embedding of $\mathbb{X}$ enacted by W, this issue would be resolved by employing only linear activation functions. However, in more general cases (such as in presence of a generic non-linear embedding) a linear activation function would not allow us to retrieve the eigenfunctions. Furthermore, the `ReLU` activation function significantly speeds up training compared to other activation functions.

3.1.b $d \gg 1$

With higher dimensions, training the model becomes more challenging due to the specific architecture required for optimal performance. Simply selecting the model that achieves the lowest loss, $\mathcal{L}$, is not sufficient. The constrained approach proposed in [5] emphasises that, in addition to attaining a low loss, the chosen architecture must also effectively avoid overfitting. Hence, a grid search should typically be performed to identify a good architecture for the specific system under investigation. In this example, the grid search was conducted by varying the depth and width of the auto-encoder, as well as the number of units in the auxiliary network. For instance, when $d = 10$, retrieving the frequencies $\omega_j,\ j = 1, \ldots, d$ is not as straightforward as it might appear with the described constrained Koopman framework; merely deepening the network relative to the described $d = 1$ case is not sufficient.

If our goal is both to obtain a low-dimensional representation of the system and to accurately extract its principal frequencies, a simple approach such as the one described in section (1.3.a) may be sufficient. Moreover, by employing an auto-encoder with linear activation functions, it is possible to reconstruct the transformation matrix W here used to embed the original state space (of dimension $2d$) into the high-dimensional space M. In this context, the Koopman operator $\mathcal{K}$ is essentially a matrix performing a linear fit between the embedded

components $\boldsymbol{\varphi}(\boldsymbol{x}_t)$, as in Eq. (1.45). Given that the chosen latent dimension is $N = 2d$, the closest approximation to an analytical linear description of the system can be obtained via a *shadow Hamiltonian* [36].

The concept of the shadow Hamiltonian originates from the study of symplectic integrators in Hamiltonian dynamics, where the discrete-time integrator exactly conserves a modified Hamiltonian that is close to the true Hamiltonian of the system. In our setting, this idea is adapted to approximate the evolution of multiple harmonic oscillators. Consider a system of harmonic oscillators described by:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{y}, \\ \dot{\boldsymbol{y}} = -\boldsymbol{\omega}^2 \boldsymbol{x}, \end{cases} \tag{3.1}$$

where $\boldsymbol{\omega}$ is a vector of natural frequencies. Discretising this system with a time step dt yields:

$$\begin{cases} \boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{y}_{t-1}\, dt, \\ \boldsymbol{y}_t = \boldsymbol{y}_{t-1} - \boldsymbol{\omega}^2 \boldsymbol{x}_{t-1}\, dt. \end{cases} \tag{3.2}$$

A refined, symplectic discretisation leads to:

$$\begin{cases} \boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{y}_t dt \\ \boldsymbol{y}_t = \boldsymbol{y}_{t-1} - \boldsymbol{\omega}^2 \boldsymbol{x}_{t-1} dt \end{cases} \implies \begin{cases} \boldsymbol{x}_t = \boldsymbol{x}_{t-1}\big(1 - \boldsymbol{\omega}^2 dt^2\big) + \boldsymbol{y}_{t-1}\, dt, \\ \boldsymbol{y}_t = \boldsymbol{y}_{t-1} - \boldsymbol{\omega}^2 \boldsymbol{x}_{t-1}\, dt, \end{cases} \tag{3.3}$$

which results in the propagation matrix

$$M_j = \begin{pmatrix} 1 - \omega_j^2 dt^2 & dt \\ -\omega_j^2 dt & 1 \end{pmatrix}, \quad \forall j = 1, \ldots, d \tag{3.4}$$

with unit determinant, i.e. $\det\big(M_j\big) = 1$. In our framework, the neural network is trained to learn an operator $\mathcal{K}$ that approximates this matrix $\boldsymbol{M}$, built with the d blocks M_j, in the embedded space. From the eigenvalues of $\boldsymbol{M}$, the frequencies ω_j of the system can then be extracted. Our results show that the model accurately reconstructs 9 out of the 10 employed frequencies. The accuracy of the reconstruction is influenced by both the chosen network architecture and the integration time step dt, which plays a crucial role in capturing the system's dynamics.

3.2 Simple model with discrete spectrum

Following the approach of [5], we implemented the proposed Koopman architecture from section (1.3.b) to solve a simple non-linear physical system with a

discrete spectrum:

$$\begin{cases} \dot{x}_1 = \mu x_1, \\ \dot{x}_2 = \gamma(x_2 - x_1^2). \end{cases} \tag{3.5}$$

This dynamical system has been extensively studied in the literature. For stable eigenvalues satisfying $\gamma < \mu < 0$, the system exhibits a slow manifold given by $x_2 = x_1^2$.

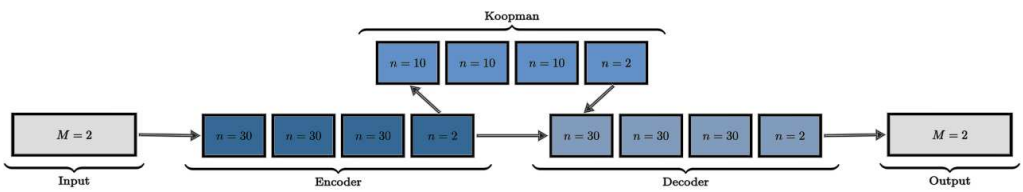

Figure 3.4: Neural Network architecture: the input and output dimensions match the latent space dimension, $N = 2$. The encoder expands the input signal through three layers of $n = 30$ neurons each and a final layer of $n = 2$ units. The decoder mirrors this structure, while the Koopman auxiliary network consists of a deep dense neural network with three layers of $n = 10$ units using `ReLU` activation function, followed by a final linear layer with $n = 2$ neurons.

In this experiment, we set $\mu_{\text{True}} = -0.05$ and $\gamma_{\text{True}} = -1$. The training dataset consists of 5×10^3 trajectories, while both the validation and test sets contain 10^3 trajectories each. Initial conditions are randomly sampled on the (x_1, x_2) plane and evolved according to Eq. (3.5) with an integration time step $dt = 0.02$ for a total length of $T = 51$ time points.

The encoder consists of three layers with 30 neurons each, using the `ReLU` activation function, except for the last linear layer with $n = 2$ units. The decoder mirrors the encoder architecture. Additionally, an auxiliary network, responsible for estimating the system's spectrum, is modelled with three layers of 10 units each, also employing the `ReLU` activation function, and a final linear layer of dimension $N = 2$.

The hyperparameter in Eq. (1.49) is set to $S_p = 30$, while the weights $\alpha_1, \alpha_2, \alpha_3$ in the total loss function (Eq. (1.50)) are set to $0.1, 1$, and 0.1, respectively, to prioritise the linear term and enhance the training of the Koopman operator in the latent space. The latter has dimension 2, as we focus on estimating the two eigenvalues μ and γ.

It is worth mentioning that, the proposed system can be fully linearised:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 - bx_1^2 \end{pmatrix} \implies \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} \mu & 0 \\ 0 & \gamma \end{pmatrix}}_{\mathcal{K}} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \tag{3.6}$$

with $b = \gamma/(\gamma - 2\mu)$. This representation is fully solvable and corresponds to the Koopman block structure in Eq. (1.51) with $\omega = 0$.

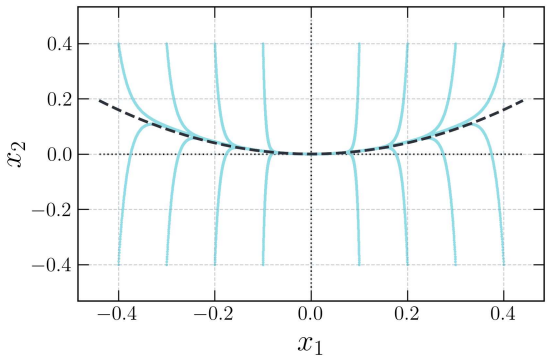

Figure 3.5: Simple non-linear system: trajectories of the system in Eq. (3.5) on the (x_1, x_2) plane, starting from different initial conditions. The dashed black line represents the invariant manifold $x_2 = x_1^2$.

Since no oscillatory behaviour is expected in this system, we can further simplify the model by considering a single Koopman block with the exponential terms $e^{\mu dt}$ and $e^{\gamma dt}$ on the diagonal.

The first two loss terms in Eq. (1.50) play a crucial role in this process. The auto-encoder is forced to extract the two latent components $\boldsymbol{y} = (y_1, y_2)^\top$ from the original data $\boldsymbol{x} = (x_1, x_2)^\top$, enabling a direct computation of the system's spectrum.

Although the system has a discrete spectrum, we did not explicitly enforce this *a priori* knowledge in our model. As illustrated in Fig. (3.6), the model naturally discovers continuous eigenvalues. Nevertheless, the continuous spectrum is tightly clustered around the discrete values $\mu = -0.05$, $\gamma = -1$. The action of the linear kernel and the coordinate linearisation are further confirmed by the transformation of the invariant parabola (depicted in black in Fig. (3.5)). While in the original system the inertial manifold is given by $x_2 = x_1^2$, in the new coordinates it is mapped to $y_2 = x_2 - bx_1^2$, effectively flattening the manifold in the transformed space $(y_1,\ y_2)$.

These results demonstrate that the combined use of auto-encoders and Koopman operators allows us to successfully retrieve the principal eigenfunctions predicted by analytical theory. We will now proceed with a more complex

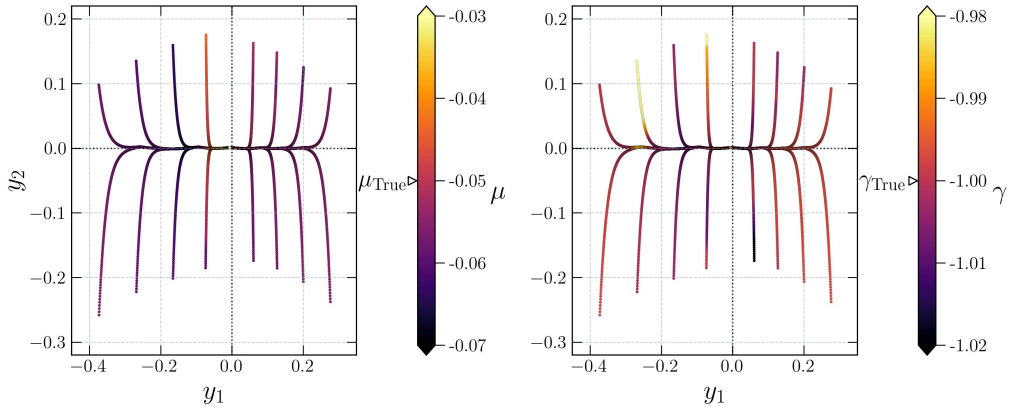

Figure 3.6: Linear embedding via the $\mathcal{K}$ operator: embedded coordinates $\boldsymbol{y}$ of the non-linear system. The plots show the estimated system eigenvalues, μ (left), γ (right), for the same trajectories in Fig. (3.5). On average, the continuous eigenvalues closely match the true ones of the original system.

system to further evaluate the capabilities of this framework.

3.3 Hopf bifurcation model

In the theory of non-linear dynamical systems[37], a Hopf bifurcation is a critical point at which a system's stability changes and a periodic solution emerges. At this bifurcation, a fixed point of a dynamical system loses stability as a pair of complex conjugate eigenvalues crosses the imaginary axis, resulting in the onset of a small-amplitude limit cycle under generic conditions. This phenomenon is observed in numerous physical models, including the Lotka-Volterra predator–prey interaction [38], the Hodgkin–Huxley model for nerve membrane potential [39], the Lorenz attractor [40], the Brusselator [41], and many more.

This behaviour is exemplified by the Stuart–Landau system:

$$\begin{cases} \dot{x} = \left(a - R^2\right)x - \omega y \\ \dot{y} = \omega x + \left(a - R^2\right)y \end{cases} \quad , \quad \text{with } R^2 = x^2 + y^2. \tag{3.7}$$

For $a - R^2 < 0$, the origin is a stable fixed point and trajectories spiral towards it. Conversely, when $a - R^2 > 0$, the origin becomes unstable and the system evolves towards a stable limit cycle with radius $r = \sqrt{a}$ and angular velocity ω.

Fig. (3.7) illustrates the subcritical and supercritical phases of the Stuart–Landau system. For the specific choice of parameters, namely $a = -0.05$, $\omega = 2$, the system exhibits subcritical behaviour, with $a - R^2 < 0$, so that trajectories converge from the initial condition $(x_0, y_0) = (0.5, 0.5)$ to the origin. In contrast, for

$a = 2.1$, $\omega = 1$ and the same initial condition, the system stabilises on a limit cycle with radius $r \approx \sqrt{a} \approx 1.45$, and as time progresses, $a - R^2$ approaches zero from above.

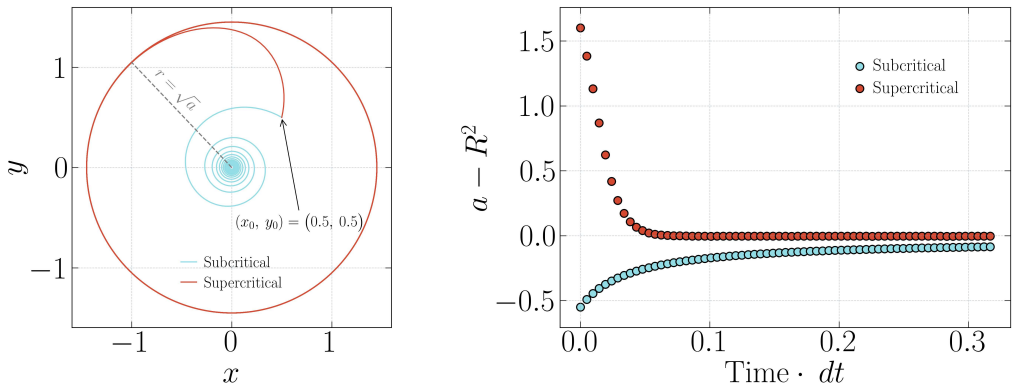

Figure 3.7: Hopf bifurcation. Left plot: subcritical (stable fixed point, $a - R^2 < 0$, with $a = -0.05$) and supercritical (limit cycle, $a - R^2 > 0$, with $a = 2.1$) phases of the Stuart–Landau system. Right plot: behaviour of the real part of the system's eigenvalues, $\lambda_\pm = (a - R^2) \pm i\omega$. As the system evolves, the real part of the eigenvalues approaches zero in the supercritical phase and -0.05 in the subcritical phase.

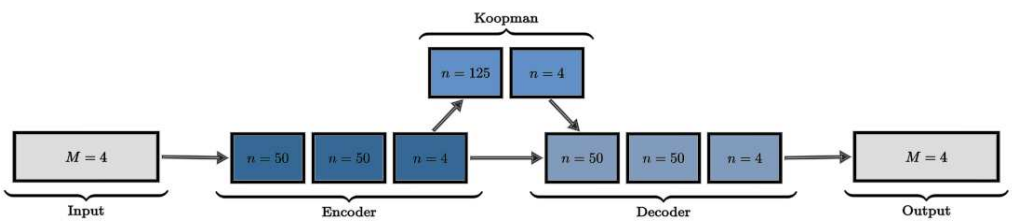

Figure 3.8: Neural Network architecture: the input and output dimensions match the latent space dimension, $N = 4$. The encoder expands the input signal through two layers of $n = 50$ neurons each, with `ReLU` activation function, and a final linear layer of $n = 4$ units. The decoder mirrors this structure, while the Koopman auxiliary network consists of a deep dense neural network with a non-linear layer of $n = 125$ units, followed by a final linear layer with $n = 4$ neurons.

The training set is composed of 10^4 trajectories, while both the validation and test sets consist of $2 \cdot 10^3$ trajectories each. All these sets include both subcritical and supercritical phases, with a and ω chosen as specified above. The integration time step is set to $dt = 0.04$, resulting in orbits of length $T = 101$ points. The hyperparameter in Eq. (1.49) is set to $S_p = 50$. The encoder consists of two layers with 50 neurons each, followed by a linear layer with 4 units. `ReLU` activation function is used in the first two layers. The decoder mirrors this structure. The auxiliary network, used to estimate the Koopman eigenvalues, comprises a single

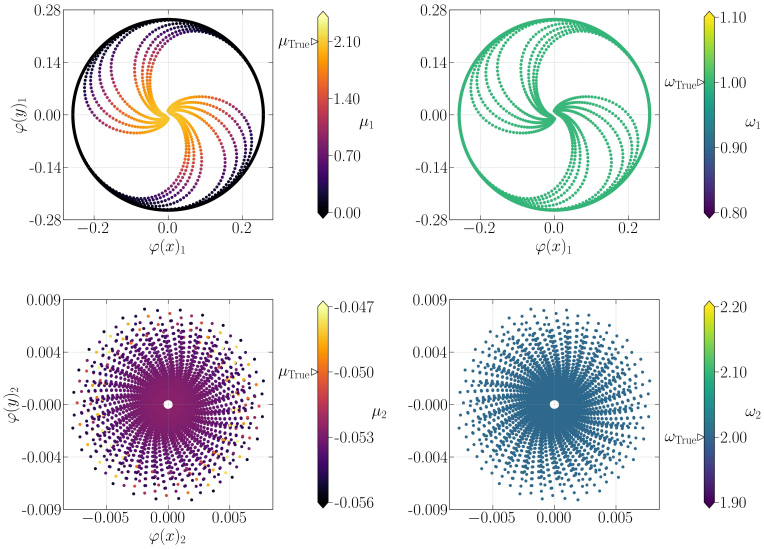

Figure 3.9: Linear embedding via the $\mathcal{K}$ operator: embedded coordinates of the Hopf bifurcation model. The top plots display the eigenvalue pair (μ, ω) for the supercritical phase, while the bottom plots show the estimated eigenvalues for the subcritical phase. The estimated values exhibit a strong agreement with the expected eigenvalues used in the simulation.

hidden layer with 125 neurons and `ReLU` activation function, followed by a linear output layer that retrieves the system eigenvalues. The architecture differs from the one presented in the previous example because network design is highly problem-dependent. In this case, the chosen architecture resulted from a grid search aimed at optimizing performance for the specific task.

Due to the presence of two oscillatory modes in the sustained and overdamped phases of the system described by Eq. (3.7), the latent space dimension was set to 4. This could allow the network to capture two pairs of eigenfunctions and eigenvalues, namely:

$$\begin{aligned}
&\big(\varphi(x)_1, \varphi(y)_1\big) \to (\mu_1, \omega_1) \quad \text{(sustained oscillations)},\\
&\big(\varphi(x)_2, \varphi(y)_2\big) \to (\mu_2, \omega_2) \quad \text{(overdamped behaviour)}.
\end{aligned}$$

Fig. (3.9) displays the results of the Koopman linear embedding. The eigenfunctions of the Koopman operator successfully capture the oscillatory patterns, with the frequencies ω_1 and ω_2 both coinciding with the correct oscillator frequency ω. Instead, the real parts μ_1, μ_2 vary continuously, and coincide with the bifurcation parameter a only when the system is near the origin. In fact, the real part of the eigenvalues coincides with $a - R^2$, which is a continuously varying quantity depending on R, a non-linear combination of x and y. The observed

behaviour of μ_1, μ_2 is in perfect agreement with the difference $\left(a - R_{\varphi}^2\right)$ shown in the right panel of Fig. (3.7). Thus, the Stuart-Landau oscillator can be captured by a simple low-dimensional combination of eigenfunctions with continuous spectrum.

Although this analysis enables us to retrieve the continuous spectrum of the initial system, its main purpose is to achieve a global linearization of the original non-linear dynamics. Unlike discrete-spectrum approaches, this framework does not require an explicit mapping into a higher-dimensional latent space. In our case, the auto-encoder was constrained to a latent dimension of 4 – matching the dimensionality of two oscillators in the original system coordinates – yet it effectively captured the system's dynamics across the entire phase space. The fact that the network fits the data well confirms that a continuous-spectrum Koopman representation is sufficient to describe the Hopf oscillator. These results align with the approach presented in [20] and further support the effectiveness of continuous-spectrum Koopman analysis.

Overall, these results confirm that a continuous-spectrum Koopman representation effectively captures the essential oscillatory modes of the Hopf system. While the network successfully linearizes the dynamics, further investigations could explore how different latent space constraints influence the representation of regions far from the fixed point.

4 Analysis of real data

In this chapter, we extend our study to real-world data, as described in chapter 2. The analysis is structured into three main sections. First, a preliminary investigation is conducted in which each patient is examined individually, allowing for a patient-specific characterization of the extracted latent dynamics. Next, a collective analysis is performed, aiming to identify common patterns and shared structures across the dataset. Finally, the chapter concludes with a discussion on the overall findings, highlighting key observations and considering alternative methods that could further enhance the analysis.

4.1 Latent dimension estimation

The first challenge is to correctly determine the appropriate low-dimensional structure of the latent manifold. Different architectures and latent space dimensions have been tested to identify the optimal neural network structure. The chosen model consists of a two-layer encoder with 256 neurons (with `ReLU` activation function) in the first layer and N neurons (with linear activation) in the second layer. The Koopman layer is implemented as an auxiliary network that processes the encoded signal through a dense layer of 300 units (with `ReLU` activation function), followed by a final linear layer that outputs the N Koopman eigenvalues $\boldsymbol{\mu}$, $\boldsymbol{\omega}$. Finally, a two-layer decoder mirrors the encoder, with 256 neurons (with `ReLU`) in the first layer and 234 neurons (matching the input dimensionality, as described in (2.2)) in the second layer, using a linear activation function.

The hyperparameters were set to $S_p = 500$, and the α_i coefficients in Eq. (1.50) were set to $0.1, 1, 0.1$, respectively. Every model was trained for at least 200 epochs, with larger models – i.e., those with higher N – trained for up to 300

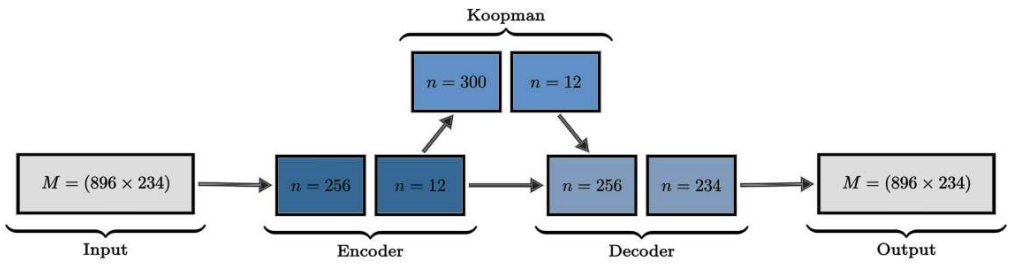

Figure 4.1: Neural network architecture: the input and output dimensions correspond to the 234 ROIs of each patient. The encoder consists of a single layer with ReLU activation function and $n = 256$ neurons, followed by a linear layer that embeds the coordinates into a latent space of dimension $N = 12$. The decoder reconstructs the input from this compressed representation back to the source space. The Koopman auxiliary network, designed to retrieve the system's eigenvalues, consists of a layer with 300 neurons and ReLU activation function, followed by a final linear layer with $N = 12$ neurons.

epochs.

To determine the appropriate latent dimensionality, we trained multiple neural networks on the entire dataset, selecting the architecture with the best reconstruction performance. To evaluate how well the model captures brain dynamics, we compared the empirical functional connectivity, $\mathrm{FC}_{\mathrm{Emp}}$, of each patient with the model-derived connectivity, $\mathrm{FC}_{\mathrm{model}}$. Functional connectivity, computed as the temporal correlation between different brain regions, results in a symmetric matrix of dimensions $(234, 234)$. The agreement between empirical and model-derived FC matrices was quantified using the total correlation ρ_{tot}, providing a single measure of the model's ability to reproduce brain-wide interactions.

Selecting the model solely based on training and validation loss is unreliable due to the risk of overfitting. The leftmost plot in Fig. (4.2) shows an elbow around $N = 12$, marking a transition between underfitted and overfitted models. In general, the optimal architecture balances good performance with a minimal number of parameters.

Another key factor in model selection is the reconstruction accuracy, ρ_{tot}. This metric, widely used in neuroscience, measures the correlation between $\mathrm{FC}_{\mathrm{Emp}}$ and the reconstructed functional connectivity from different models ($\mathrm{FC}_{\text{auto-encoder}}$, $\mathrm{FC}_{\mathrm{Koopman}}$). We first evaluated a standard auto-encoder, replicating the analysis in [42], and then extended it to the Koopman-augmented model.

Although ρ_{tot} provides a straightforward assessment of the model's ability to capture brain dynamics, it does not directly evaluate the quality of the learned latent representation or the dynamical consistency of the reconstructed trajectories. Nevertheless, it remains the standard metric for FC-based models. Notably,

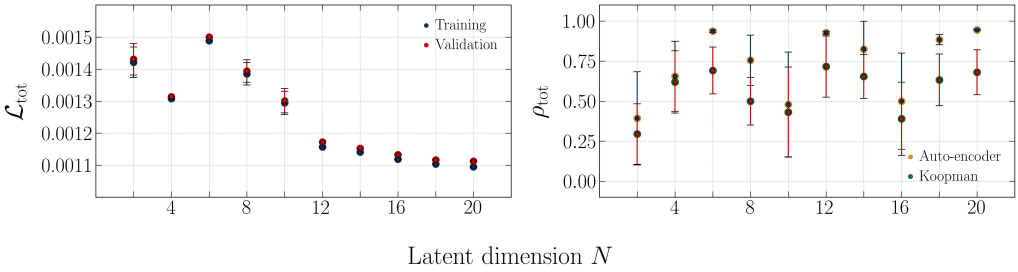

Figure 4.2: Optimal latent dimension N identification: training and validation loss (left) and ρ_{tot} (right) plotted against increasing latent dimension N. Error bars represent the standard deviation computed over 5 independent runs of the same system.

ρ_{tot} is maximized at $N = 12$, where the performance gap between the auto-encoder and Koopman models is also minimized.

These findings suggest that $N = 12$ is a suitable estimate of the system's latent dimensionality, consistent with [42], where $N = 6$ was reported for a standard auto-encoder on the same dataset.

4.2 Per-subject analysis

Starting from the idea that each patient carries both general and individual information about their condition and the severity of the damage, we analyse each subject independently. The overarching goal is to characterize each patient by their own continuous spectrum, described by the Koopman eigenvalues $\big(\mu(t),\ \omega(t)\big)$, as previously demonstrated in section (3.3).

Thus, we expect each patient to exhibit distinct eigenvalues and possibly a different latent representation. In contrast, healthy control subjects should display similar spectra, as their dynamics is not expected to vary significantly.

The use of functional connectivity as a validation metric for computational models of brain dynamics is well established in neuroscience. It has been widely employed to assess how well simulated brain activity captures large-scale patterns of synchronization observed in real data [43, 44]. Honey et al. (2009) demonstrated that large-scale FC patterns can be predicted from underlying structural connectivity and dynamical models, while Deco et al. (2011) further highlighted the importance of comparing empirical and simulated FC to assess model validity. This approach has since been applied in various studies exploring whole-brain modelling, including those based on data-driven and machine learning methods [45, 46].

By leveraging this methodology, we ensure that the Koopman-based neural network not only captures individual latent representations but also maintains

temporal correlations between different brain regions.

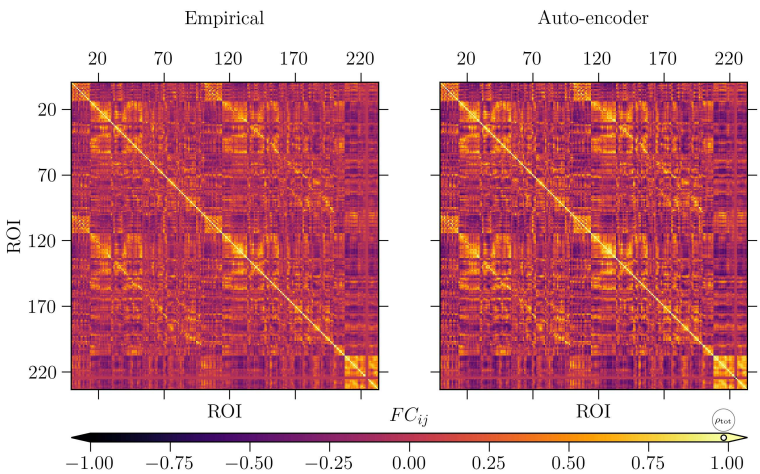

Figure 4.3: Empirical and auto-encoder FC comparison for a healthy subject (training set). $\rho_{\text{tot}} = 0.985$ represents the global correlation between the two matrices.

A crucial challenge in a single-patient approach is the appropriate division of the dataset into training and validation sets. Various methods exist, such as time-windowing and bootstrap resampling techniques [47], which segment fMRI data into overlapping or shuffled time windows to enhance generalization. However, since our study focuses on preserving the temporal structure of the data, we opted for a straightforward division: the first 700 time points constitute the training set, while the remaining 196 form the validation set. This approach ensures that the model learns from the past while predicting future states.

Due to the stochastic nature of resting-state fMRI, the correlation between training and validation sets is inherently limited. This is quantified by ρ_{max}, the empirical upper bound on correlation, which remains strictly below 1 because of neural activity fluctuations introducing variability that the model cannot perfectly capture.

Fig. (4.3) shows that the auto-encoder accurately reconstructs the empirical functional connectivity for the training set, achieving near-perfect correlation: $\rho_{\text{tot}} = 0.985$.

On the validation set, performance decreases, yielding $\rho_{\text{tot}} = 0.421$ with $\rho_{\text{max}} = 0.563$. This confirms that even an ideal model cannot exceed the intrinsic correlation ρ_{max} between training and validation data, as resting-state fMRI exhibits structured temporal dependencies that constrain the achievable agreement.

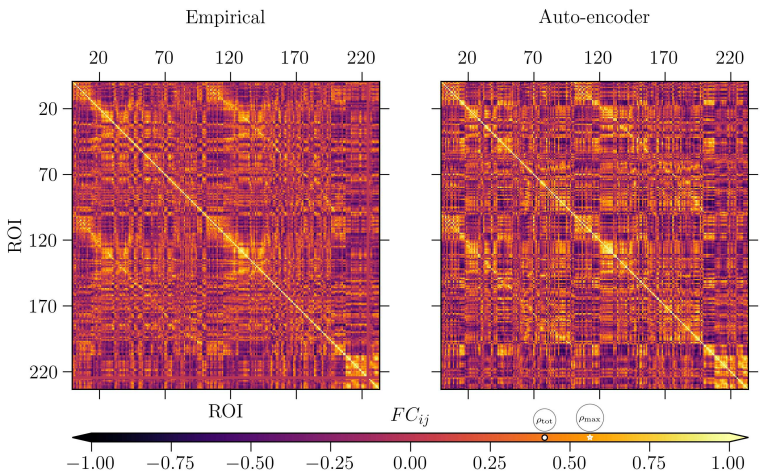

Figure 4.4: Comparison between empirical and auto-encoder-derived FC for a healthy subject (validation set). The global correlation between the two matrices is $\rho_{\text{tot}} = 0.421$, while $\rho_{\text{max}} = 0.563$ represents the empirical correlation between the training and validation sets for the shown subject.

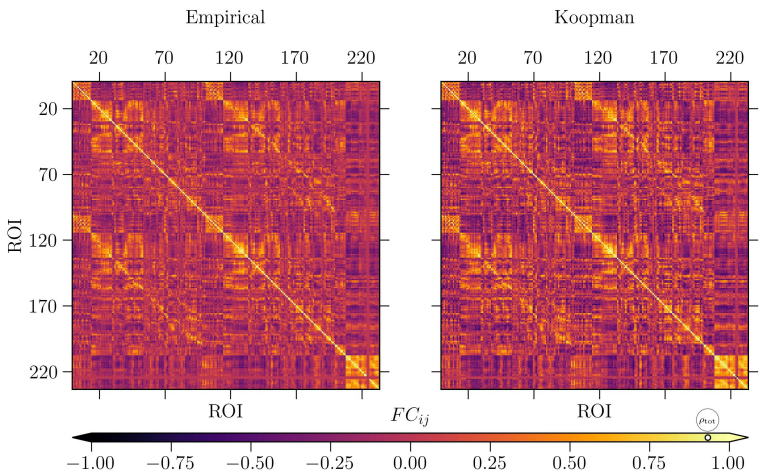

Figure 4.5: Empirical and Koopman FC comparison for a healthy subject (training set). The global correlation between the two matrices is $\rho_{\text{tot}} = 0.932$.

Next, we introduced the auxiliary network between the encoder and decoder layers to estimate the Koopman eigenvalues. Fig. (4.5) demonstrates the predictive capabilities of this full Koopman-based network, achieving a global correlation of $\rho_{\text{tot}} = 0.932$ for the presented subject.

While the total correlation remains high, it is slightly lower than in the simple auto-encoder model. The validation performance, however, degrades signifi-

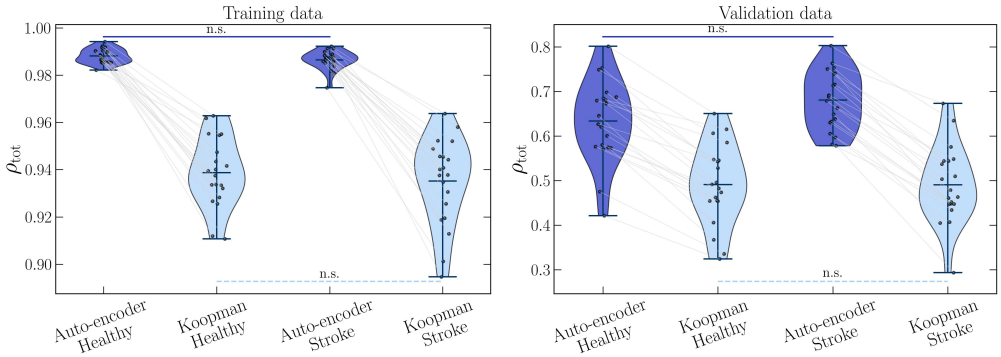

Figure 4.6: Correlation coefficients ρ_{tot} for training (left) and validation (right) sets. Grey lines connect results for the same subjects across models.

cantly, indicating that the Koopman formulation does not enhance predictive accuracy.

Fig. (4.6) compares ρ_{tot} for the auto-encoder and Koopman model, comparing healthy subjects with stroke patients. Overall, validation performance is lower than training, and the enforcement of a linear dynamics in latent space via the Koopman operator reduces reconstruction accuracy.

It is important to note that the analysis does not easily allow classifying subjects as healthy controls or stroke patients. Specifically, inspecting the estimated Koopman eigenvalues reveals no clearing differentiation between the two groups.

However, statistical testing reveals subtle differences, as in Fig. (4.7). The μ distributions exhibit a distinct shape, with all values being negative. This suggests that all latent modes correspond to overdamped oscillators. The distributions show a wider spread, particularly in the range $[-0.04, -0.02]\ s^{-1}$. Moreover, healthy controls reach lower μ values compared to stroke patients, and exhibit a denser distribution around slightly higher μ values compared to stroke patients. On the other hand, the ω distribution for stroke patients extends to higher frequency values and experience altered Koopman frequency patterns compared to healthy subjects one.

A two-sided Student's t-test confirms these significant distinctions:

$$t_\nu^\mu = -25.8, \qquad p_\mu < 10^{-10},$$
$$t_\nu^\omega = -6.3, \qquad p_\omega < 10^{-10},$$

with $\nu = 167998$ degrees of freedom.

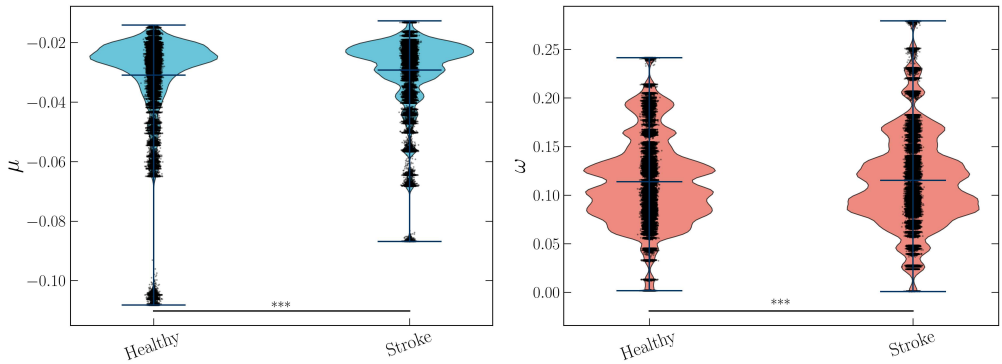

Figure 4.7: Distribution of Koopman eigenvalues $\lambda_j(t) = \mu_j(t) \pm i\omega_j(t)$, $j = 1, \ldots, 12$, $t = 1, \ldots, T$: real part μ (left) and imaginary part ω (right) in the training set. A significant correlation was found between the distributions ($p_{\text{value}} < 0.001$).

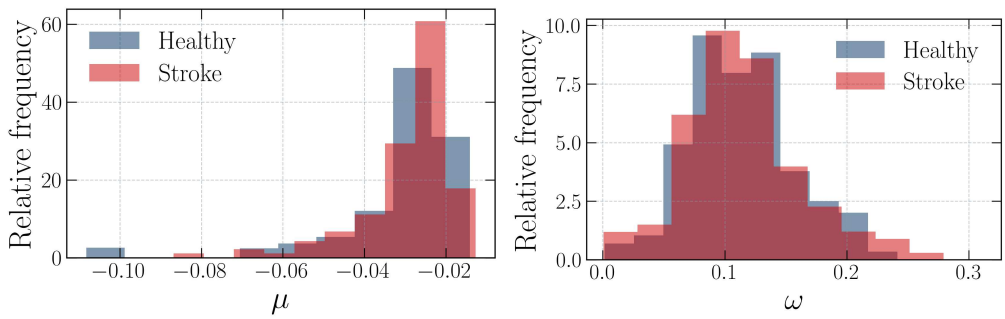

Figure 4.8: Histograms of Koopman eigenvalues: real part μ (left) and imaginary part ω (right) in the training set.

These statistics aggregate eigenvalues across both temporal ($t = 1, \ldots, T$) and dimensional indices ($j = 1, \ldots, 12$).

Healthy controls exhibit broader μ distributions but narrower ω distributions compared to stroke patients.

μ values might be related to the system's ability to maintain stable dynamics over time, with stroke patients displaying greater dispersion and lower values, possibly indicating impaired stability. ω may reflect variations in oscillatory patterns, with differences likely attributable to altered brain dynamics following a stroke.

4.3 All-subjects analysis

While subject-specific training enhances embedding precision and predictive accuracy, it complicates eigenvalue comparisons across subjects due to potential

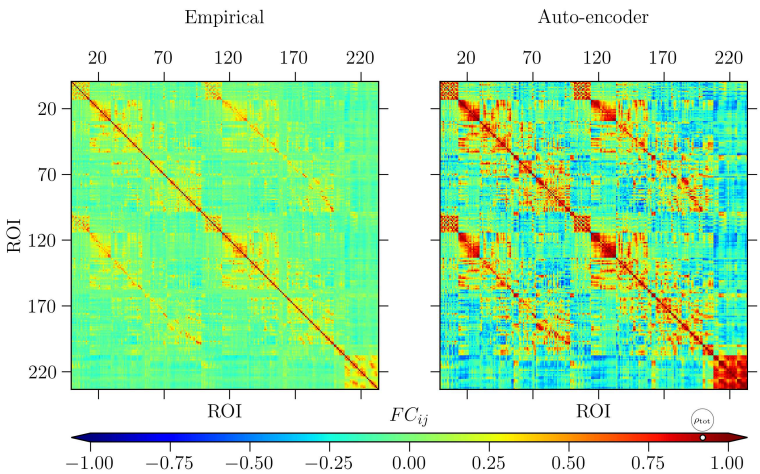

Figure 4.9: All-subject training set comparison between empirical and auto-encoder FC. $\rho_{\text{tot}} = 0.919$ represents the global correlation between the two matrices.

differences in learned latent representations. Additionally, training-validation splitting remains non-trivial. To address these concerns, we now transition to training a single neural network incorporating information from all subjects, both healthy controls and stroke patients.

Training all subjects together has the advantage of retrieving a unique neural network that generalizes the dataset's properties. Moreover, this approach ensures that the embedded latent space is shared among all subjects, allowing for a direct comparison between modes of different subjects. Furthermore, there is now a natural way of splitting the dataset into training and validation sets. The split can now be performed at the subject level, considering 80% of the subjects as the training set – 98 subjects in total – and the remaining 20% – 25 subjects – as the validation set. In this manner, the neural network takes as input the whole information pertaining to specific subjects, i.e., the described matrix of shape (896×234), and seeks a common eigenspace along with the corresponding eigenvalues of the Koopman operator.

The main drawback of this approach is the increased difficulty in designing a network that can simultaneously capture both population-level features and individual subject characteristics. This poses a challenge in learning a latent representation that encodes both the general properties of the dataset and the subject-specific continuous spectra.

Using the same architecture described in the previous section, we assessed both the reconstruction and prediction capabilities of the trained network.

As Fig. (4.9) illustrates, the pure auto-encoder model is able to reconstruct

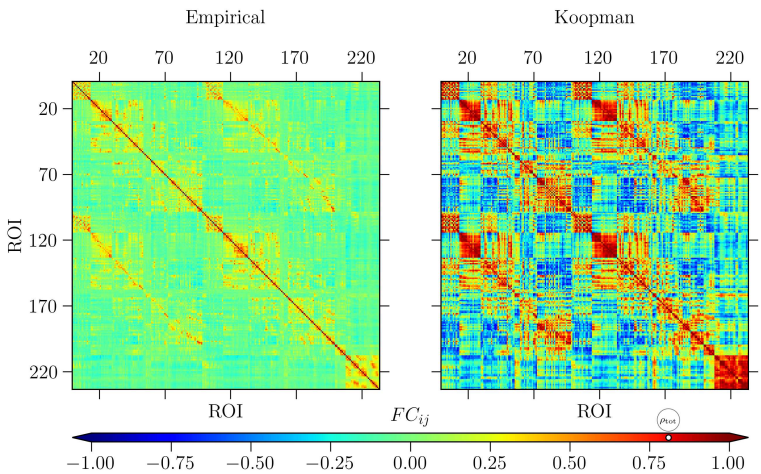

Figure 4.10: All-subject training set comparison between empirical and Koopman FC. $\rho_{\text{tot}} = 0.809$ represents the global correlation between the two matrices.

the system's correlations, identifying the main communities within the system. However, the reconstructed correlations are exaggerated, in the sense that empirical positive (negative) correlations appear more positive (negative) in the auto-encoder reconstruction.

The same conclusion holds for the full model, including the Koopman operator. The total correlation, $\rho_{\text{tot}} = 0.809$, obtained is lower than the one retrieved with the auto-encoder alone.

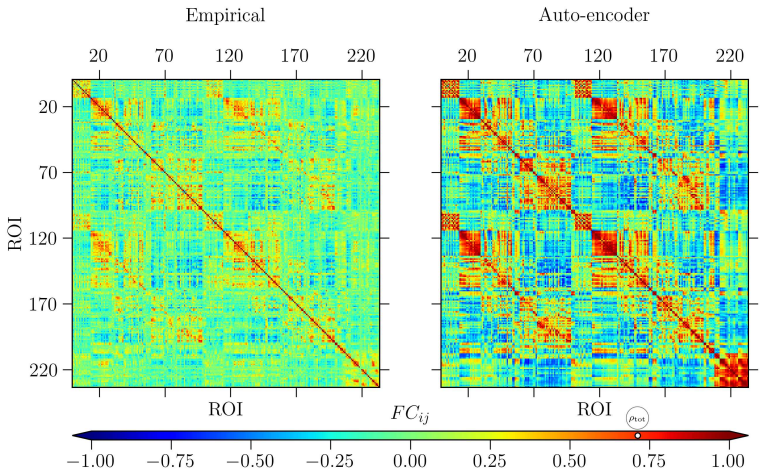

Figure 4.11: Same subject shown in Fig. (4.3) from the training set: comparison between empirical and auto-encoder FC. $\rho_{\text{tot}} = 0.712$ represents the global correlation between the two matrices. Its value is lower than the one retrieved in the previous analysis.

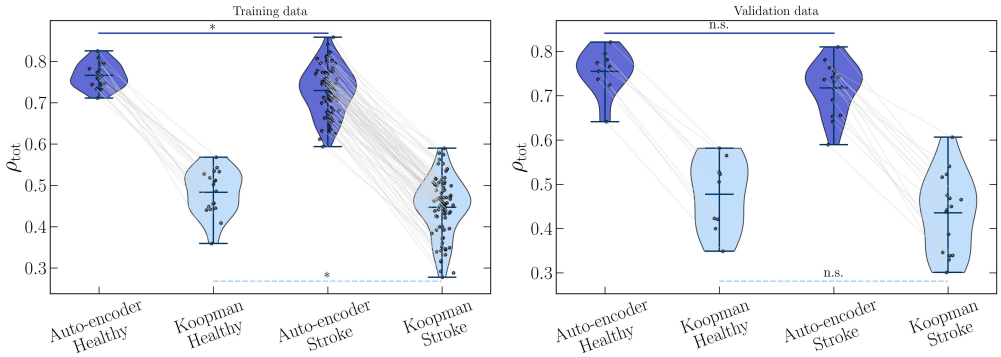

Figure 4.12: Correlation coefficients ρ_{tot} for training (left) and validation (right) sets. Grey lines connect results for the same subjects across models.

The results obtained on the validation set closely match those of the training set, demonstrating good generalization capabilities and significantly improving the poor results obtained in the previous section: $\rho_{\text{tot}} = 0.910$ for the auto-encoder and $\rho_{\text{tot}} = 0.789$ for the Koopman model.

However, the presented results refer to the dataset as a whole. At the individual subject level, both reconstruction accuracy and correlation with empirical data drop significantly.

Fig. (4.11) presents the same subject, specifically from the healthy control group, as shown in Fig. (4.3). This direct comparison highlights the difficulty of the trained network in specializing at a single subject level, leading to a global correlation coefficient $\rho_{\text{tot}} = 0.712$, more than 25% lower than the one previously retrieved. As illustrated in Fig. (4.12), the Koopman model generally achieves a worse reconstruction compared to the auto-encoder.

Although a statistically significant difference is found between healthy controls and patients in both models, with

$$t_\nu^{\text{auto-enc}} = 2.6, \qquad p_{\text{auto-enc}} = 0.01,$$
$$t_\nu^{\text{Koopman}} = 2.1, \qquad p_{\text{Koopman}} = 0.04,$$

with $\nu = 96$ degrees of freedom, this difference is no longer significant in the validation set (right plot, $p > 0.1$, with $\nu = 23$ degrees of freedom).

The same can be said for the eigenvalues comparison between the two groups. As already shown in Fig. (4.7), the split between healthy controls and patients must be known *a priori*, as it cannot be retrieved through a data-driven dynamical analysis. Notably, the distributions exhibit a multimodal structure, with approximately three peaks for μ and four for ω, which suggests the presence of distinct dynamical modes. These peaks appear within the expected frequency

band, aligning with the physiological range of brain oscillations.

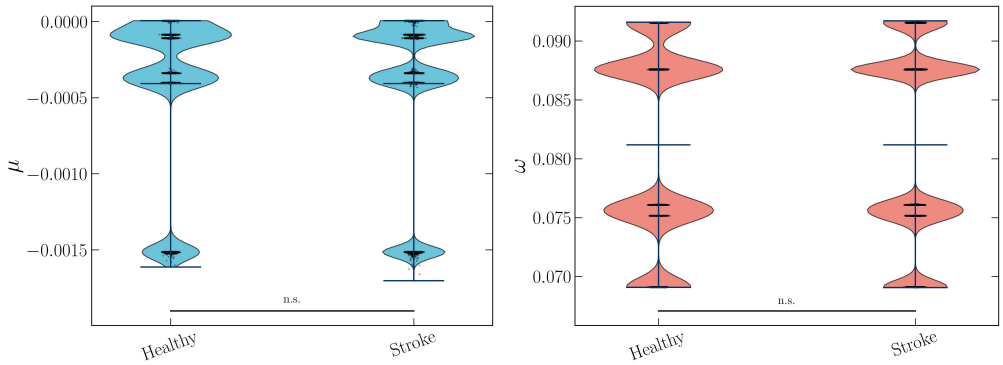

Figure 4.13: Distribution of Koopman eigenvalues $\lambda_j(t) = \mu_j(t) \pm i\omega_j(t)$, $j = 1, \ldots, 12$, $t = 1, \ldots, T$: real part μ (left) and imaginary part ω (right) in the training set.

The neural network generalizes well across subjects, capturing consistent patterns in both healthy and stroke populations. However, it does not seem to specialize in distinguishing pathological conditions, as no significant differences emerge between the two groups. This could indicate that the Koopman operator framework successfully learns global latent dynamics but does not explicitly separate healthy from pathological states.

Additionally, the construction of these distinct modes remains unclear. The emergence of specific peaks suggests that the network identifies recurrent dynamical structures, but their exact origin, whether driven by underlying neurophysiological mechanisms or artifacts of the learning process, requires further investigation.

4.3.a Homotopic connectivity

Homotopic connectivity refers to the functional and structural connections between corresponding regions in the left and right hemispheres of the brain. These inter-hemispheric connections are crucial for integrating information processed separately in each hemisphere, thereby facilitating coherent cognitive and motor functions. Studies have demonstrated that homotopic regions exhibit strong functional connectivity, which is essential for maintaining balanced and coordinated brain activity.

Research indicates that the strength of homotopic functional connectivity (HoFC) can influence intra-hemispheric functional integration. Specifically, variations in HoFC have been associated with differences in functional integration

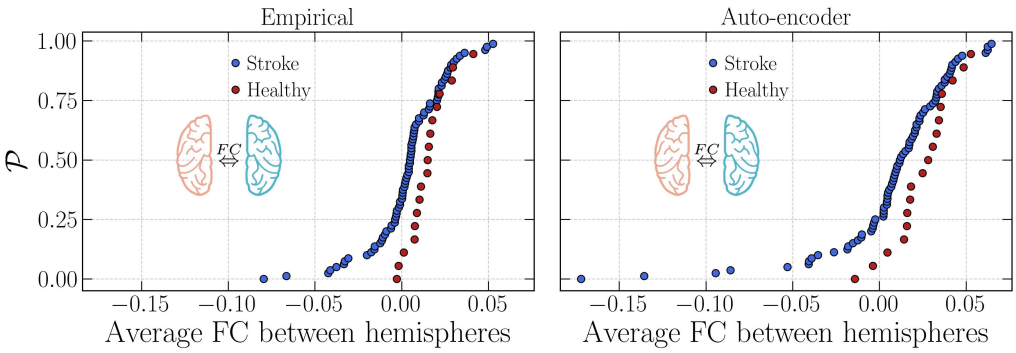

Figure 4.14: Inter-hemispheric correlation cumulative distribution: comparison between stroke patients and healthy controls. Empirical original data (left plot) and auto-encoder reconstruction (right plot).

within each hemisphere, suggesting that inter-hemispheric interactions play a role in modulating lateralized brain functions [48].

Moreover, disruptions in homotopic connectivity have been linked to various neurological conditions. For instance, alterations in HoFC patterns have been observed in patients with stroke, where changes in inter-hemispheric coherence correlate with behavioural deficits. This highlights the significance of homotopic connections in both healthy brain function and in the context of brain injuries [49].

Understanding the dynamics of homotopic connectivity not only sheds light on the fundamental organization of the brain but also provides insights into the mechanisms underlying functional lateralization and the impact of inter-hemispheric interactions on cognitive processes.

We analysed the mean functional connectivity between homotopic brain regions, focusing on cortical regions (thus excluding the 34 subcortical ROIs). This analysis was performed on the empirical data as well as on the reconstructions provided by the auto-encoder and Koopman models[1].

In Fig. (4.14) we plot the cumulative distribution function of the mean inter-hemispheric correlation for both healthy subjects and stroke patients. As expected, a clear difference emerges. Notably, this distinction is preserved in the auto-encoder model, which, despite amplifying negative correlations for some patients, maintains a clear separation between the two groups.

However, this distinction is entirely lost when applying the full Koopman model.

[1]GRsohn's brain image, under Creative Commons Attribution License (CC BY 3.0).

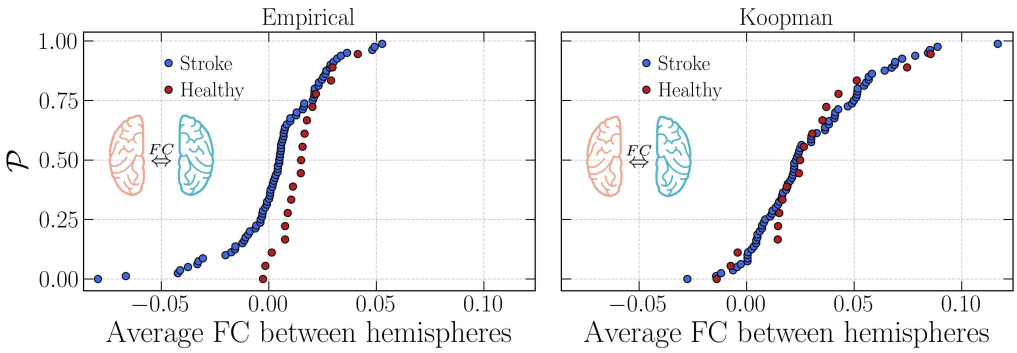

Figure 4.15: Inter-hemispheric correlation cumulative distribution: comparison between stroke patients and healthy controls. Empirical original data (left plot) and Koopman reconstruction (right plot).

This result highlights the difficulty of the Koopman-based model in preserving individual features. Although the latent space is the same, from the Koopman-predicted spectrum is not possible to differentiate the members of the two groups. A potential explanation for this behaviour is overfitting, which may be mitigated by exploring higher latent dimensions N.

4.4 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique that transforms a high-dimensional dataset into a lower-dimensional space while preserving as much variance as possible [50, 51]. Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of observations and d the number of features, PCA finds a new set of orthogonal basis vectors, called principal components, that maximize the variance of the projected data.

Mathematically, PCA is performed by computing the covariance matrix of the zero-centred data:

$$\mathbf{\Sigma} = \frac{1}{n}\left(\mathbf{X}^\top \mathbf{X}\right). \tag{4.1}$$

The principal components are obtained by solving the eigenvalue problem:

$$\mathbf{\Sigma}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \tag{4.2}$$

where λ_i are the eigenvalues and $\mathbf{v}_i$ the corresponding eigenvectors of $\mathbf{\Sigma}$. The eigenvectors, sorted in descending order by their eigenvalues, form the transformation matrix $\mathbf{V} \in \mathbb{R}^{d \times d}$. The first k eigenvectors define the subspace onto which the data is projected:

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_k, \tag{4.3}$$

where $\mathbf{V}_k$ contains the first k principal components, and $\mathbf{Z} \in \mathbb{R}^{n \times k}$ represents the transformed data in the reduced space.

PCA is particularly useful for noise reduction, feature selection, and visualization of high-dimensional datasets. In the context of brain connectivity analysis, it has been employed to extract dominant patterns of functional connectivity and compare them across different conditions [52]. The choice of the optimal number of components k is typically guided by the cumulative explained variance:

$$\text{Cumulative Explained Variance} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}, \tag{4.4}$$

which quantifies how much variance each principal component captures relative to the total variance. This metric allows identifying the minimum number of principal components needed to retain a desired fraction of the total variance, often using a threshold of 95% or 99%.

In this section, we employed PCA as an alternative method for evaluating the latent representations obtained through the auto-encoder-Koopman framework, enabling a direct comparison between linear and non-linear dimensionality reduction techniques.

4.4.a PCA results

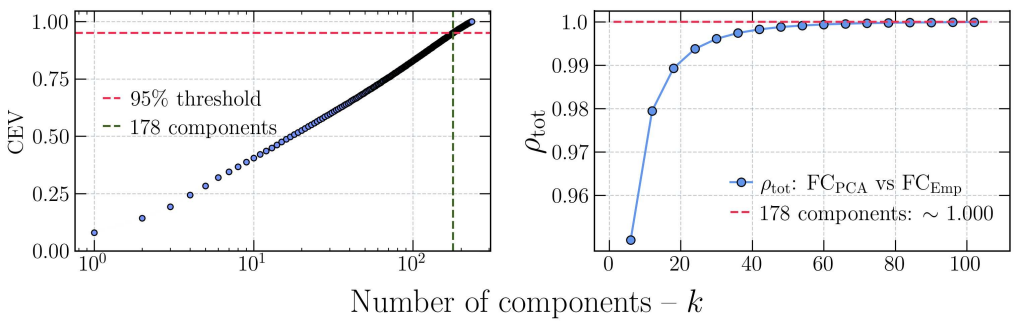

Figure 4.16: Cumulative Explained Variance (left plot): 178 principal components are required to explain 95% of the dataset variance (x-axis in logarithmic scale). Total correlation coefficient ρ_{tot} between the FC of the original dataset, FC_{Emp}, and the reconstruction via PCA, FC_{PCA} (right plot).

We started by selecting $k = 12$ principal components. The total correlation between the empirical FC of all participants, FC_{Emp}, and the reconstructed one

after the PCA reduction, FC_{PCA}, is 0.979, higher than that obtained with the auto-encoder, 0.919, and the one obtained including a Koopman layer, 0.809. The mean squared error between the original data and their reconstructed version, through these three different methods, is small ($\sim 10^{-3}$), with PCA yielding the smallest value.

If we include the PCA results in Fig. (4.6), we see that PCA achieves a higher ρ_{tot} and there is no statistically significant difference between healthy controls and stroke participants.

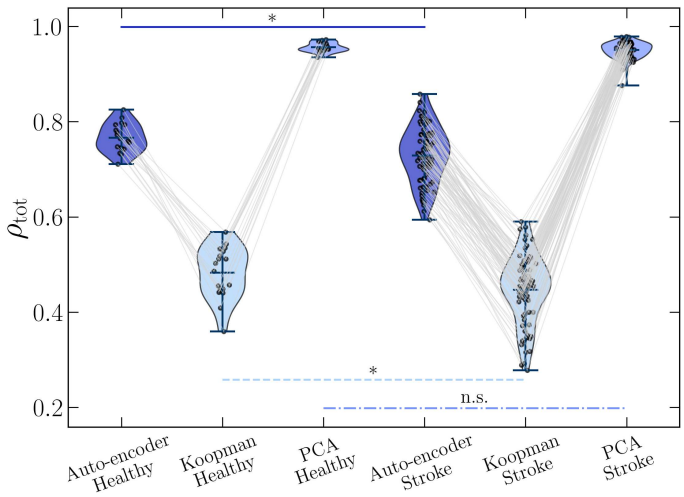

Figure 4.17: Correlation coefficients ρ_{tot} for the training set. Grey lines connect results for the same subjects across different models: auto-encoder, Koopman, and PCA for healthy and stroke subjects.

The high ρ_{tot} in Fig. (4.17) can be explained by the fact that PCA is a linear technique that, even with only 12 components, preserves the principal directions of dataset variability. However, if we look at the Cumulative Explained Variance, as defined in Eq. (4.4), we see that a total of $N = 178$ components are required to explain 95% of the dataset variance.

The fact that, even with $k = 6$ components, PCA achieves a total correlation $\rho_{\text{tot}}^{k=6} > 0.95$ shows that FC is dominated by a low-dimensional set of principal components. *A posteriori*, this would suggest complementing ρ_{tot} with subtler metrics that could capture other aspects of the dynamics.

Finally, the dimension $k = 178$, the one thanks to which PCA is able to explain 95% of the dataset variance, could potentially serve as an estimate of the optimal dimensionality of the Koopman operator, helping to mitigate overfitting when training the neural network.

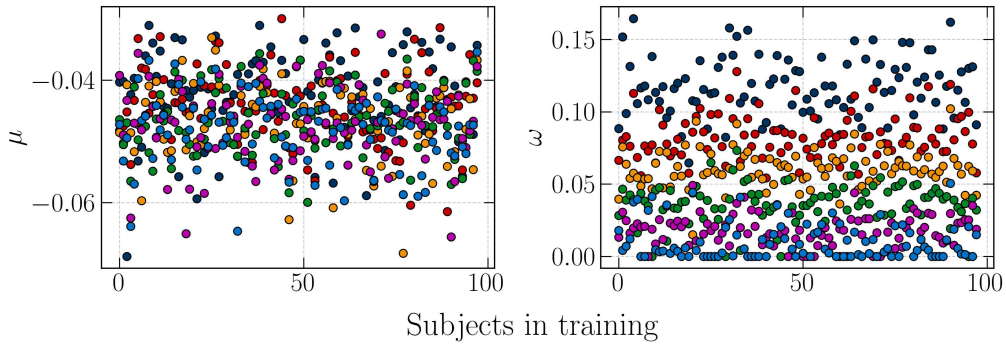

Figure 4.18: PCA eigenvalues mapped into the Koopman block structure: distribution of μ (left) and ω (right) values across subjects.

Moreover, following the idea of employing a linear model, we performed a linear fit of the time-evolved embedded components obtained through PCA. Specifically, for each patient, we considered the principal components extracted from their data and fitted a linear regression model of the form described in Eq. (1.45), where each time step was used to predict the next one. This approach allowed us to assess the degree of dynamical linearity of the PCA-reduced representation. We observed that the principal components identified by this method exhibit a higher degree of linearity than expected. The mean R^2 coefficient for patients in the training set is 0.880 for $k = 178$ and 0.834 for $k = 12$.

The eigenvalues obtained via this linear fitting are more scattered than those found via the Koopman operator.

When mapping the eigenvalues $\Lambda_j,\ j = 1, \ldots, k$, obtained by PCA into the complex-valued representation $\lambda_j = \mu_j \pm i\omega_j$, as expected from the block structure in Eq. (1.51), we observe that PCA eigenvalues exhibit a broader spread across the complex plane compared to those from the Koopman framework (Fig. (4.13)) that, instead, are more tightly clustered within a narrower range. This suggests that the Koopman approach imposes additional constraints on the latent dynamics, potentially enforcing a more structured representation of the system's evolution. However, the comparison between the eigenvalue distributions does not reveal statistically significant differences between healthy controls and stroke patients (results not shown).

In combination with this fully linear approach – PCA reduction and linear fitting of the embedded components – we also attempted a non-linear modelling of the principal components identified by PCA, by training a fully connected neural network. This approach aims to mimic the flow action, as in Eq. (1.3), but

applied to the k principal components $\boldsymbol{y} \in \mathcal{V} \subset \mathbb{R}^{T \times k}$:

$$\boldsymbol{y}(\boldsymbol{x}_{t+1}) = \boldsymbol{\Phi}\big(\boldsymbol{y}(\boldsymbol{x}_t)\big). \tag{4.5}$$

The neural network is trained to approximate the mapping $\boldsymbol{\Phi} : \mathcal{V} \to \mathcal{V}$ through non-linear activation functions. However, this approach did not yield satisfactory results, as the learned transformation did not exhibit the expected linear trends in the latent space. Even after testing multiple neural network architectures, the model failed to produce stable eigenvalues and meaningful temporal predictions, suggesting that the learned representation did not effectively capture the system's dynamics.

4.5 Conclusions

In this thesis, we have explored a data-driven embedding approach for the analysis and linearisation of complex dynamical systems through the lens of Koopman operator theory. Our work used both synthetic data, from simulated dynamical systems, and a real fMRI stroke dataset, enabling us to assess the strengths and limitations of the proposed methodologies under controlled and real-world conditions.

4.5.a Synthetic Data Analysis

Using synthetic data, we demonstrated that a properly designed auto-encoder can recover the essential dynamical modes of a system, approximating the Koopman operator. In particular, we showed that for systems with a discrete spectrum, such as the harmonic oscillator or the simple model introduced in section (3.2), the network was able to extract key eigenfunctions and eigenvalues that faithfully represent the underlying dynamics. The analyses on these synthetic systems highlighted that the neural network could achieve a global linearisation, albeit with some challenges in regions far from the fixed point. These results validate the feasibility of using a Koopman-based approach for capturing the system's dynamics in cases where a discrete spectral decomposition is appropriate. Furthermore, we extended this analysis to systems exhibiting a continuous spectrum, such as the Hopf oscillator, showing that the Koopman framework can also provide meaningful representations in this setting, although with different interpretative challenges.

4.5.b Analysis of Real Data

The application of the proposed framework to a real stroke dataset further illustrated both its potential and its limitations. We conducted two types of analyses: a per-subject analysis and a pooled analysis where all subjects (both healthy controls and stroke patients) were processed by a single neural network.

Training the model on individual subjects allowed for a more precise embedding and improved predictive accuracy. This approach effectively preserved subject-specific functional connectivity patterns, ensuring that the reconstructed dynamics closely matched the empirical data. However, this setup introduced a critical limitation: differences in the learned latent representations made it difficult to compare eigenvalues across subjects in a meaningful way. Additionally, due to the very limited data available for each subject, it posed a challenge for model generalization.

To address these issues, we explored an alternative strategy: training a single neural network on all subjects simultaneously, incorporating both healthy controls and stroke patients. In the pooled analysis, the model successfully captured global, population-level features of the dataset, effectively learning a common latent structure shared among subjects. However, this came at the cost of losing subject-specific details. Rather than maintaining individual functional connectivity patterns, the model tended to homogenize dynamical differences in the latent space. As a consequence, key distinctions between healthy subjects and stroke patients became less pronounced, and the model's ability to retain unique individual characteristics was significantly reduced. This limitation is particularly evident in the reconstruction of functional connectivity, where the correlation with empirical data declined compared to the per-subject approach. These results suggest that, while pooling data improves the robustness of the learned Koopman representations, it risks missing the variability inherent in stroke data, ultimately blurring critical subject-specific information.

4.5.c Model Selection and Latent Space Considerations

An important aspect of our study was the determination of an appropriate latent space dimension. We performed extensive grid searches, varying the depth and width of the auto-encoder as well as the number of units in the auxiliary network. Our analysis, using criteria such as the total reconstruction accuracy ρ_{tot} and the model loss $\mathcal{L}$, indicated that a latent dimension of $N = 12$ offered a good trade-off between model complexity and performance. Nevertheless, these metrics also revealed that while the auto-encoder performs well in reconstructing functional connectivity, including the Koopman operator imposes additional

constraints that may sometimes limit the model's ability to capture subject-specific variations.

4.5.d Implications and Future Directions

The findings of this thesis provide considerable insight into the application of Koopman operator theory for the linearisation of non-linear dynamical systems. First, our results on synthetic data confirmed that data-driven embedding methods, particularly those based on auto-encoders, can successfully approximate the Koopman operator, thereby allowing for a global linearisation of complex dynamics. Second, our analyses on real fMRI data highlighted a crucial trade-off: while a continuous-spectrum Koopman representation can robustly capture the main oscillatory modes of the system, it may also reduce the network's sensitivity to inter-subject differences. This is especially problematic in clinical datasets, where individual variability is of paramount importance.

These observations motivate several avenues for future research. One promising direction is to explore more flexible latent representations that can accommodate both global linearisation and subject-specific variability. Additionally, integrating prior knowledge about the system (e.g., known structural connectivity or physiological constraints) could further refine the model's ability to capture both general and individual features.

Finally, while the present study has focused on the application of Koopman operator theory to resting-state fMRI data, the underlying framework is broadly applicable to a wide range of complex dynamical systems. Future work could investigate its extension to other domains, such as financial time series, ecological models, or engineering systems, where an analytical description is not easily available.

In summary, this thesis has assessed the potential of combining deep learning techniques with Koopman operator theory to achieve a data-driven linearisation of complex systems. Despite some limitations in generalising individual subject characteristics, the proposed framework represents a significant step towards developing predictive models that are both interpretable and robust. Our findings suggest that with further refinement, such hybrid models could play a pivotal role in understanding and controlling the dynamics of systems where traditional analytical approaches fall short.

Bibliography

[1] Bernard O Koopman. "Hamiltonian systems and transformation in Hilbert space". In: *Proceedings of the National Academy of Sciences* 17.5 (1931), pp. 315–318.

[2] Igor Mezić and Andrzej Banaszuk. "Comparison of systems with complex behavior". In: *Physica D: Nonlinear Phenomena* 197.1-2 (2004), pp. 101–133.

[3] Igor Mezić. "Spectral properties of dynamical systems, model reduction and decompositions". In: *Nonlinear Dynamics* 41 (2005), pp. 309–325.

[4] Petar Bevanda, Stefan Sosnowski, and Sandra Hirche. "Koopman operator dynamical models: Learning, analysis and control". In: *Annual Reviews in Control* 52 (2021), pp. 197–212.

[5] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. "Deep learning for universal linear embeddings of nonlinear dynamics". In: *Nature communications* 9.1 (2018), p. 4950.

[6] Peter J Schmid. "Dynamic mode decomposition of numerical and experimental data". In: *Journal of fluid mechanics* 656 (2010), pp. 5–28.

[7] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. "Dynamic mode decomposition with control". In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161.

[8] J v Neumann. "Physical applications of the ergodic hypothesis". In: *Proceedings of the National Academy of Sciences* 18.3 (1932), pp. 263–266.

[9] George D Birkhoff. "Proof of the ergodic theorem". In: *Proceedings of the National Academy of Sciences* 17.12 (1931), pp. 656–660.

[10] Milan Korda and Igor Mezić. "Optimal construction of Koopman eigenfunctions for prediction and control". In: *IEEE Transactions on Automatic Control* 65.12 (2020), pp. 5114–5129.

[11] Yoshihiko Susuki and Igor Mezić. "A prony approximation of Koopman mode decomposition". In: *2015 54th IEEE conference on decision and control (CDC)*. IEEE. 2015, pp. 7022–7027.

[12] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[13] Qianxiao Li et al. "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.10 (2017).

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[15] David H Hubel and Torsten N Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of physiology* 160.1 (1962), p. 106.

[16] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[18] Balázs Csanád Csáji et al. "Approximation with artificial neural networks". In: *Faculty of Sciences, Etvs Lornd University, Hungary* 24.48 (2001), p. 7.

[19] Mark A Kramer. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* 37.2 (1991), pp. 233–243.

[20] Chufan Li and Yueheng Lan. "Koopman analysis of nonlinear systems with a neural network representation". In: *Communications in Theoretical Physics* 74.9 (2022), p. 095604.

[21] Natasza Marrouch et al. "Data-driven Koopman operator approach for computational neuroscience". In: *Annals of Mathematics and Artificial Intelligence* 88.11 (2020), pp. 1155–1173.

[22] Tanja Eisner et al. *Operator theoretic aspects of ergodic theory*. Vol. 272. Springer, 2015.

[23] Seth M Hirsh et al. "Structured time-delay models for dynamical systems with connections to Frenet–Serret frame". In: *Proceedings of the Royal Society A* 477.2254 (2021), p. 20210097.

[24] Maurizio Corbetta et al. "Common behavioral clusters and subcortical anatomy in stroke". In: *Neuron* 85.5 (2015), pp. 927–941.

[25] Chiara Favaretto et al. "Subcortical-cortical dynamical states of the human brain and their breakdown in stroke". In: *Nature communications* 13.1 (2022), p. 5069.

[26] Seiji Ogawa et al. "Brain magnetic resonance imaging with contrast dependent on blood oxygenation." In: *proceedings of the National Academy of Sciences* 87.24 (1990), pp. 9868–9872.

[27] Peter A Bandettini et al. "Time course EPI of human brain function during task activation". In: *Magnetic resonance in medicine* 25.2 (1992), pp. 390–397.

[28] SA Huettel. "Functional magnetic resonance imaging". In: *Sinauer associate* (2004).

[29] Nikos K Logothetis. "What we can do and what we cannot do with fMRI". In: *Nature* 453.7197 (2008), pp. 869–878.

[30] Stephen M Smith et al. "Functional connectomics from resting-state fMRI". In: *Trends in cognitive sciences* 17.12 (2013), pp. 666–682.

[31] Joshua Sarfaty Siegel et al. "Disruptions of network connectivity predict impairment in multiple behavioral domains after stroke". In: *Proceedings of the National Academy of Sciences* 113.30 (2016), E4367–E4376.

[32] Joshua S Siegel et al. "Re-emergence of modular brain networks in stroke recovery". In: *Cortex* 101 (2018), pp. 44–59.

[33] Alexander Schaefer et al. "Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity MRI". In: *Cerebral cortex* 28.9 (2018), pp. 3095–3114.

[34] Nathalie Tzourio-Mazoyer et al. "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain". In: *Neuroimage* 15.1 (2002), pp. 273–289.

[35] Fredrik Gustafsson. "Determining the initial states in forward-backward filtering". In: *IEEE Transactions on signal processing* 44.4 (1996), pp. 988–992.

[36] Benedict Leimkuhler and Sebastian Reich. *Simulating hamiltonian dynamics*. 14. Cambridge university press, 2004.

[37] John Michael Tutill Thompson and H Bruce Stewart. *Nonlinear dynamics and chaos*. John Wiley & Sons, 2002.

[38] Peter J Wangersky. "Lotka-Volterra population models". In: *Annual Review of Ecology and Systematics* 9 (1978), pp. 189–218.

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

[40] Edward N Lorenz. "Deterministic nonperiodic flow". In: *Journal of atmospheric sciences* 20.2 (1963), pp. 130–141.

[41] John J Tyson. "Some further studies of nonlinear oscillations in chemical systems". In: *The Journal of Chemical Physics* 58.9 (1973), pp. 3919–3930.

[42] Sebastian Idesis et al. "A low dimensional embedding of brain dynamics enhances diagnostic accuracy and behavioral prediction in stroke". In: *Scientific Reports* 13.1 (2023), p. 15698.

[43] Gustavo Deco, Viktor K Jirsa, and Anthony R McIntosh. "Emerging concepts for the dynamical organization of resting-state activity in the brain". In: *Nature reviews neuroscience* 12.1 (2011), pp. 43–56.

[44] Christopher J Honey et al. "Predicting human resting-state functional connectivity from structural connectivity". In: *Proceedings of the National Academy of Sciences* 106.6 (2009), pp. 2035–2040.

[45] Morten L Kringelbach et al. "Dynamic coupling of whole-brain neuronal and neurotransmitter systems". In: *Proceedings of the National Academy of Sciences* 117.17 (2020), pp. 9566–9576.

[46] Joana Cabral et al. "Cognitive performance in healthy older adults relates to spontaneous switching between states of functional connectivity during rest". In: *Scientific reports* 7.1 (2017), p. 5135.

[47] Pierre Bellec, Vincent Perlbarg, and Alan C Evans. "Bootstrap generation and evaluation of an fMRI simulation database". In: *Magnetic resonance imaging* 27.10 (2009), pp. 1382–1396.

[48] JH Martínez et al. "Role of inter-hemispheric connections in functional brain networks". In: *Scientific reports* 8.1 (2018), p. 10246.

[49] Xinhu Jin, Xinyu Liang, and Gaolang Gong. "The relationship between interhemispheric homotopic functional connectivity and left-right difference of intrahemispheric functional integration in the human brain". In: *Imaging Neuroscience* 2 (2024), pp. 1–18.

[50] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[51] Harold Hotelling. "Analysis of a complex of statistical variables into principal components". In: *Journal of Educational Psychology* 24.6 (1933), p. 417.

[52] Martin J. McKeown et al. "Analysis of fMRI data by blind separation into independent spatial components". In: *Human Brain Mapping* 6.3 (1998), pp. 160–188.