Robust and Interpretable Learning for Operator-Theoretic Modeling of Non-linear Dynamics

by

Shaowu Pan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2021

Doctoral Committee:

Associate Professor Karthik Duraisamy, Chair
Assistant Professor Alex Gorodetsky
Professor Nathan Kutz, University of Washington
Professor Venkat Raman
Assistant Professor Aaron Towne

Shaowu Pan

shawnpan@umich.edu

ORCID iD: 0000-0002-2462-362X

For my parents.

ACKNOWLEDGEMENTS

First, I want to sincerely appreciate my advisor, Professor Karthik Duraisamy. His guidance over academics has a profound influence on my research development. Karthik provided me with an incredible amount of freedom to pursue my research ideas and external collaborations. Besides the conventional one-on-one bi-weekly meetings, we had countless great conversations over Slack on sharing the latest papers, discussing interesting ideas and perspectives in the research directions for the past 4.5 years ranging from 7 AM to 1 AM. His attitude of always thinking out of "fluids" and diving into papers that are not in our "comfort zone" has a significant impact on me. Besides, as one of the best academic speakers I had ever seen, Karthik helped me greatly in writing every single paper and preparation for every single presentation, including faculty job interview. His high standard of optimally delivering research outcomes has a huge influence on me. I feel incredibly fortunate to have such a mentor and wish I can emulate his example.

Second, I want to thank all my committee members: Venkat Raman, Aaron Towne, Alex Gorodetsky, and Nathan Kutz, for their insightful comments and useful suggestions on not only my thesis but also my research career in the future. Moreover, I also want to deeply thank my M.S. advisor, Professor Eric Johnsen, for his strong recommendation so that I could get admitted into the Ph.D. program. I want to give special thanks to my former labmate Dr. Mauro Rodriguez for his time in going through my personal statement and resume for my Ph.D. application. I am also grateful to many faculty members at Michigan who taught me a wide range of

exciting and useful courses.

The Ph.D. journey was not easy for me. Unlike traditional fields such as CFD where the cause of failure from students is usually due to a "bug", there were many times that things did not work in data-driven modeling for physical system that is because of some fundamental issue that none of us knew at that time (e.g., we asked a wrong question). At that time, I was expecting to solve the ultimate turbulence modeling problem with machine learning when I started. As a result, the first year of transitioning from a traditional field on the first floor of FXB is the most challenging time for me. Fortunately, the fantastic atmosphere among students at CASLAB, especially the 1060 office, helped me to relax. The weekly (sometimes daily) hang out in the summer was fun with Eric, Anand, David, Helen, Nick, Chris, Yaser, Asitav, Ayoub and Adam at Jolly Pumpkin, HopCat, and Bill's Garden. We had many interesting discussions, from fluid mechanics, reduced-order modeling to cultural differences and politics. Also, the Indian food hangout at Cardamom, Authentikka, Neehee with Vishal and Aniruddhe was unforgettable. I learned most of my Hindi over the table. The biceps competition and Korean fried chicken night with Davodi and David were also fun. In addition, I wish to acknowledge all of the staff at FXB who help to create such an enjoyable community.

Also, I want to thank all my external collaborators: Jianxun Wang (Norte Dame), Weiqi Ji (MIT), Wenlong Ma (Caltech), Qi Gao (ZJU) for the numerous discussions on data-driven engineering in biomedical flows, combustion, biped robotics, and experimental fluids together with many fruitful research outcomes. In particular, I want to thank all of my good friends at Michigan, including Xiuyuan Yang, Sicheng He, Daning Huang, Jiayang Xu, Xingyu Li, Nan Li, Yan Zhao, Guodong Chen. They deserve special attention for their long-time company from the time when I was in ME to my final defense in AE, sharing viewpoints in academics and life. Also, I give special thanks to my high school classmate, Ke Yang. I had no intent to come to

the US for graduate school until he came to Beijing for a GRE training course in the summer of 2011 and asked me for a vacancy in my dormitory. I had no better things to do that summer, so I registered for that 200$ course as well. At that time, I couldn't figure out why so many students are willing to prepare for an expensive (and painful) English test in order to study and live on the other side of the planet. Now I would say that is a lobe-switching event of the chaotic Lorenz system, which is caused by the intermittent forcing on certain high-dimensional linear dynamics in the viewpoint of Koopman. I am glad everything worked out in this unexpected path.

Lastly, I want to express my greatest gratitude towards my beloved parents for their overseas support and encouragement. Unlike most international students' parents, my parents are very conservative and never get too excited about me going to a foreign country for post-graduate education. The only word in the thesis that they could understand might be my name. But still, they would have the biggest smile on their faces and have it print out, keeping it on the shelf. This Ph.D. journey wouldn't be possible without their continuous support and encouragement.

The research in this thesis were supported by DARPA under the grant titled *Physics Inspired Learning and Learning the Order and Structure Of Physics,* (Technical Monitor: Dr. Jim Gimlett), and the US Air Force Office of Scientific Research through the Center of Excellence Grant FA9550-17-1-0195 (Technical Monitors: Mitat Birkan & Fariba Fahroo).

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Non-linear dynamical systems are of significant interest to a wide range of science and engineering communities. This dissertation is focused on the advancement of theory and algorithms for operator-theoretic modeling and decomposition of non-linear dynamical systems, with a particular emphasis on the Koopman operator. The Koopman operator represents non-linear dynamics in the form of an infinite-dimensional linear operator over the space of observables of the system state. Despite the broad appeal of the Koopman operator in modal analysis, reduced-order modeling and control, discovering accurate finite-dimensional representations presents considerable challenges. Most of the existing data-driven approaches for non-linear systems target prediction, which in effect amounts to interpolation within parameter space. In contrast, the Koopman operator presents the potential of a systematic framework for physics-consistent, stable predictions, control and modal analysis.

Data-driven methods including time-delay dynamic mode decomposition (TD-DMD), extended DMD (EDMD), kernel DMD (KDMD), and deep learning-based techniques have been developed for Koopman approximations. While the promise of these techniques is clear, stability, robustness, spurious modes, and uncertainty quantification continue to be a challenge. As a consequence, the above methods are often employed as qualitative postprocessing tools instead of quantitative, robust, and accurate a posteriori models. This dissertation aims to address the above challenges, and demonstrates applications in modal decomposition, predictive modeling, and control of non-linear dynamics.

We begin by developing theoretical results on the structure of TD-DMD models

of non-linear dynamics on an attractor. We demonstrate that the minimal number of time delays required for perfect reconstruction is directly related to the sparsity of the Fourier spectrum of the dynamics. Furthermore, we explain why TD-DMD can "extrapolate in time", i.e., why a model trained on a partial period of data can perfectly predict the future. We also prove that an increase in the number of time delays benefits numerical conditioning, making the model robust to noisy data. For example, we demonstrate the numerical stabilization effect of "over-delays" on the 3D Turbulent Rayleigh-Bénard convection. We also construct a ROM based on TD-DMD, and compare it to state-of-the-art methods for a chaotic single injector combustion process.

Next, we develop robust and accurate mode selection algorithms for non-linear Koopman approximation methods (e.g., EDMD/KDMD). We propose a model-agnostic sparsity promoting framework based on a) pruning Koopman eigenfunctions that excessively deviate from linear evolution, and b) multi-task learning for selecting the most informative Koopman modes. From an analytical perspective, we show a close relationship between the well-known sparsity-promoting DMD (spDMD), and an empirical criterion through the lens of optimization. The performance of the proposed framework is demonstrated in several unsteady flow cases ranging from strongly transient flows including the cylinder wake and a 3D turbulent ship air-wake.

Finally, we propose a probabilistic deep learning framework for the continuous-time Koopman operator, with an option to either purely rely on the data, or to use the knowledge of the physics in the form of governing equations. With regard to imposing physical consistency, we introduce a novel parameterization that guarantees temporal stability of the Koopman operator. The effectiveness of the proposed framework is evaluated on several non-linear dynamical systems with varying amounts of training data and noise levels. The proposed framework shows better robustness to noise and improved accuracy compared to the standard Long Short-Term Memory (LSTM)

model. We take the unforced Duffing oscillator as an example to reveal that linear reconstruction can be useful for systems with symmetrical attractors. Lastly, we show an application of our framework on the data-driven optimal control of a simple nonlinear dynamical system, and highlight performance benefits over traditional local linearization.

In summary, this dissertation places an emphasis on the development of interpretable and robust methods for approximating the Koopman operator, and provides mathematical insight where possible.

CHAPTER I

Introduction

> We choose to go to the moon. We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard, because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one which we intend to win, and the others, too.
>
> *John F. Kennedy Moon Speech, Rice Stadium (1962)*

1.1 High Dimensional Non-linear Dynamical Systems

The need for faster, more efficient and versatile air transport vehicles continues to drive challenging design problems in Aerospace engineering. As examples, the desire

to reduce travel time from New York City to London by half has led to silent supersonic commercial aircraft initiated by *Aerion Supersonic*, while *SpaceX* is developing vehicles for space exploration as shown in fig. 1.1.

Figure 1.1: Left: Aerion SBJ designed by jet builder Aerion Supersonic that expects to fly silent supersonic planes by 2024, unlocking a $40 billion market (*Commons*, 2020a). Right: Human launch of SpaceX's Falcon 9 rocket raises the company value to $44 billion (*Commons*, 2020b).

Such design problems involve complex high-dimensional nonlinear phenomena such as turbulent combustion, laminar-to-turbulence transition, shock wave-boundary layer interaction. Despite the discovery of the Navier–Stokes equations two centuries ago, our understanding of underlying flow mechanisms and the development of efficient predictive models remains a challenge, primarily due to the non-linear, multi-scale and chaotic nature of the partial differential equations (PDE).

With the ever-increasing compute power and advances in computational algorithms, high-fidelity simulations (e.g., Direct Numerical Simulation (DNS) with one trillion unknowns on the K computer (*Ishihara et al.*, 2016)) are providing a wealth of data that would be challenging or impossible to acquire in experiments (*Bulat et al.*, 2013; *Raman et al.*, 2005). However, high fidelity simulations such as DNS become impractical in practical flow configurations due to the disparate range of time and

length scales. For typical high fidelity simulations of the Navier-stokes equations, multi-scale phenomena associated with turbulence lead to a $O(Re^{2.25})$ degrees of freedom, where Re is the Reynolds number. For example, a typical Re in an automotive application is $O(10^6)$, which requires a DNS mesh size on the order of $O(10^{13.5})$, and thus, even storing a single snapshot of such field requires 760 TB. Even for large eddy simulation (LES) which is of moderate fidelity, *Chapman* (1979) wall-resolved LES for an airfoil would require $O(Re^{1.8})$ DOF, which is still close to DNS scaling and the corresponding single snapshot at $O(10^6)$ requires 1.52 TB. It should be noted that with various "near-wall modeling" techniques, ideally speaking one could resolve only to the outer layer which reduces the scaling to $O(Re^{0.4})$ (*Chapman*, 1979). However, wall-modeling in LES is still an active research area.

In addition to multi-scale phenomena, coupling with multi-physics, e.g., chemistry, plasma, droplets and bubbles, fluid dynamics with additional non-linearity becomes even more challenging to analyze. Thus, lower fidelity methodologies have been more popular in design applications. For example, multidisciplinary design optimization (MDO) typically relies on Euler (*Mader et al.*, 2008; *He et al.*, 2019) or Reynolds-averaged Navier-Stokes equations (*Lyu et al.*, 2013). However, the modeling error introduced in these model can be significant in many situations and should be accounted for. Therefore, high-fidelity simulations or experiments are still required to validate or improve the downstream applications of these lower-fidelity models (e.g. in aerodynamic design.)

As a result, the conventional computational design paradigm faces several major challenges. First, it involves prohibitively high computational costs and human effort in the design & validation process, or in pursuing accurate models for missing physics. Second, the complex flow field from high-fidelity simulations or experiments is challenging to interpret (e.g. extracting meaningful coherent structures). Third, data from high-fidelity simulations and experiments are not fully exploited (e.g. under-

standing the flow phenomena beyond validation for low-fidelity models). Traditional theories and tools of modeling and analysis of non-linear dynamical system - originating from the controls community - are designed either for linear (*Juang and Pappa*, 1985; *Phan et al.*, 1993) or non-linear ordinary differential equations (ODE) (*Billings*, 2013; *Khalil and Grizzle*, 2002) with relatively small degrees of freedoms (DOF). Moreover, effective controllers require a computationally feasible model for online implementation and efficient solvers to minimize the latency introduced in the estimation of the states (*Proctor et al.*, 2018). Thus, the prevailing state of the art in dynamics and control requires new developments when analyzing, modeling and controlling high dimensional fluid dynamical systems.

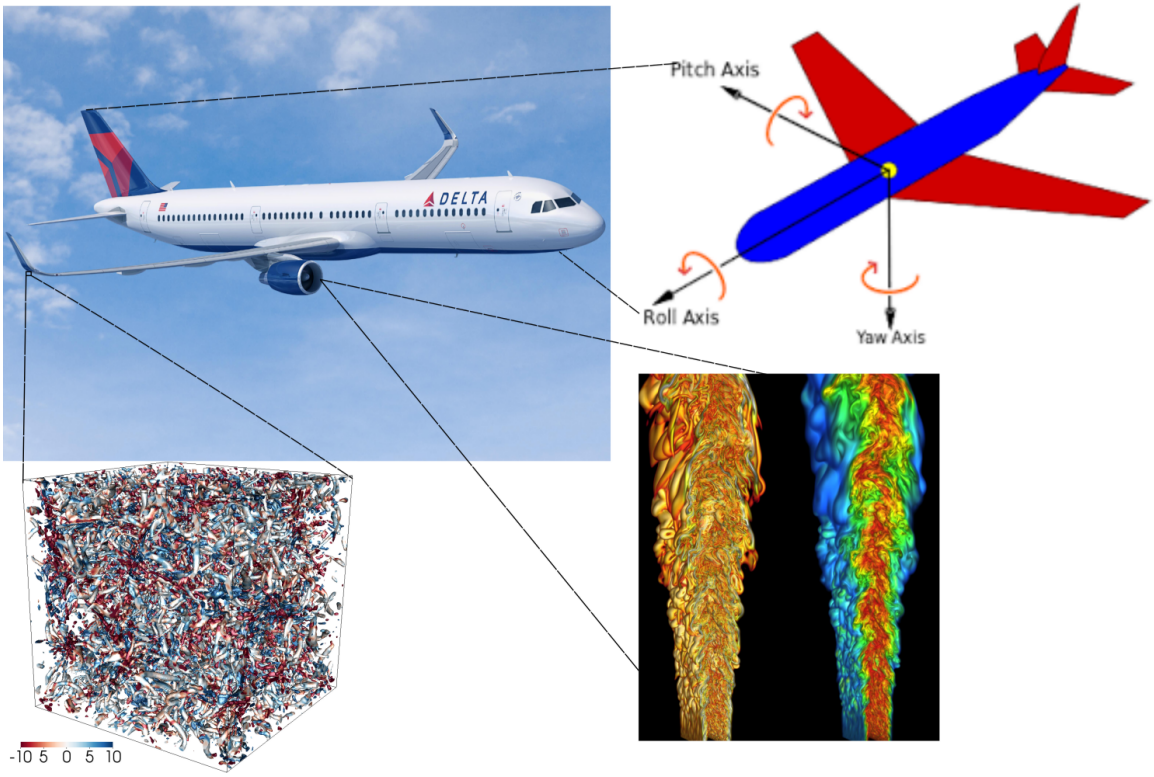


Figure 1.2: Prevalence of high dimensional non-linear systems in aerospace engineering (*Pan and Johnsen*, 2017; *Yu*, 2012).

1.2 Fluid Dynamic Modeling

The development of effective descriptive and predictive models has been actively pursued throughout the history of fluid mechanics (*Kundu and Cohen*, 2001). Successful examples include potential flow theory, waves propagation models in compressible flow and hydrodynamic instability analysis. Successful theories demonstrate clear causality between assumptions and observations. Thus, as long as the underlying assumptions are satisfied, one can safely "transfer" conclusions to many other situations. However, with the ever-more complex applications, assumptions that underpin simple theories often become too restrictive. For that reason, fluid dynamicists have leveraged inductive reasoning by searching for phenomenological assumptions from experiments that makes closed-form design feasible (*Anderson Jr*, 2010). Modern *inductive reasoning* approaches, e.g., the search for useful patterns in scientific data automatically using mathematical methods (*Brunton and Kutz*, 2019), sometimes referred to as "data-driven science and engineering", has received attention in recent years. One of the key factors for the surge is the abundant spatial-temporal data from numerical simulations e.g. DNS and LES (*Moin and Mahesh*, 1998; *Pitsch*, 2006), and experiments e.g., Tomographic PIV (*Scarano*, 2012).

This dissertation investigates *Operator theoretic learning*, with a focus on Koopman operator theory and falls in the intersection among: 1) modal analysis/decomposition of fluid flows, 2) model order reduction and flow control, 3) non-intrusive reduced order modeling. The Koopman operator provides an elegant framework that can enable a wide spectrum of applications in *data-driven modeling, prediction*, *analysis*, and *optimal control* of high-dimensional non-linear dynamical systems.

1.3 Dimensionality-reduction

The main idea of modal decomposition is to extract coherent modes, e.g., traveling waves, that capture essential features of complex dynamics. The resulting modes can help extract a low-dimensional form of the complex nonlinear flows either from an energy perspective, linearized dynamics or input-output standpoint. Based on how much information is required from the governing equation, there are data-based and operator-based techniques (*Taira et al.*, 2017). Data-based techniques obtain modal decomposition from data while operator-based techniques target the characterization of underlying operators.

For example, proper orthogonal decomposition (POD) (*Berkooz et al.*, 1993) is a data-based technique. It requires a data matrix that consists snapshots of a field of quantities of interest, e.g., velocity, pressure. The decomposition represents a low dimensional representation that optimally preserves the variance (e.g., kinetic energy for velocity). For example, consider the full field is velocity field u, v, w on a mesh of $N_x \times N_y \times N_z$ grid points sampled from $t_1, \ldots, t_M$, we can have a r-dimensional presentation of the snapshots $\mathbf{X}$,

$$\mathbf{X} = \begin{bmatrix} u(x_1, y_1, z_1, t_1) & \ldots & u(x_1, y_1, z_1, t_M) \\ v(x_1, y_1, z_1, t_1) & \ldots & v(x_1, y_1, z_1, t_M) \\ w(x_1, y_1, z_1, t_1) & \ldots & w(x_1, y_1, z_1, t_M) \\ \vdots & \vdots & \vdots \\ u(x_{N_x}, y_{N_y}, z_{N_z}, t_1) & \ldots & u(x_1, y_1, z_1, t_M) \\ v(x_{N_x}, y_{N_y}, z_{N_z}, t_1) & \ldots & v(x_1, y_1, z_1, t_M) \\ w(x_{N_x}, y_{N_y}, z_{N_z}, t_1) & \ldots & w(x_1, y_1, z_1, t_M) \end{bmatrix} \approx \sum_{1}^{r} \sigma_i \mathbf{U}_i \mathbf{V}_i^{\top}, \tag{1.1}$$

where $\mathbf{U}_i$ is the i-th POD mode and $\sigma_i \mathbf{V}_i$ is the i-th POD coefficient. Since the corresponding kinetic energy is ordered by $\sigma_1 \geqslant \sigma_2 \ldots \geqslant \sigma_r$, we have an energy-based

hierarchical description of the full field.

A closely related technique is called spectral POD (SPOD) (*Towne et al.*, 2018) in which one performs POD in the Fourier space. The resulting SPOD modes are both orthogonal to each other in time and space while the conventional POD only guarantees space orthogonality.

Another example is Balanced POD (BPOD) (*Rowley*, 2005) which is an approximation to a classic model order reduction technique called *balanced truncation*. It starts with a potentially high-dimensional linear time-invariant input-output system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{1.2}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \tag{1.3}$$

where even saving the Gramians for large fluid system is not possible. *Rowley* (2005) proposed a trade-off by additionally performing simulations with pulse inputs to the above linear system and the following adjoint system:

$$\dot{\mathbf{z}} = \mathbf{A}^\top \mathbf{z} + \mathbf{C}^\top \mathbf{v}, \tag{1.4}$$

$$\mathbf{w} = \mathbf{B}^\top \mathbf{z}. \tag{1.5}$$

One can obtain the balanced transformation by taking an SVD of the product between the response from linear system and that from the adjoint system. In terms of modal analysis, BPOD provides insight into the flow pattern that is most sensitive to control inputs and those that trigger large growth in the observables.

Examples of operator-based techniques include global linear stability analysis (*Monkewitz et al.*, 1993), which consider eigen-decompositions of the linearized system of equations (e.g. linearized Navier–Stokes equations). The resulting eigenvector represents a linearly evolving pattern following a certain growth rate and frequency. Other examples include resolvent analysis (*McKeon and Sharma*, 2010), which relates the

quadratic non-linear forcing in N-S with the linear counterpart. This starts with the full N-S equations, but separating non-linear terms on the right hand side. So that one can view the behavior of the fluid system from an input-output point of view:

$$\mathbf{M}\frac{\partial \mathbf{q}}{\partial t} + \mathbf{A}\mathbf{q} = \mathbf{f}(\mathbf{q}), \tag{1.6}$$

where $\mathbf{q}$ represents the spatially discretized flow variables (e.g., velocity, pressure). For statistically stationary flows, the corresponding frequency representation of the above becomes

$$(i\omega\mathbf{M} + \mathbf{A})\hat{\mathbf{q}} = \hat{\mathbf{f}}, \tag{1.7}$$

where $\mathbf{q}(t) = \hat{\mathbf{q}}e^{i\omega t}$, $\mathbf{f}(t) = \hat{\mathbf{f}}e^{i\omega t}$. Thus, the resolvent operator describes the amplification from non-linear interaction of different time scales in frequency space as

$$\mathbf{R} = (i\omega\mathbf{M} + \mathbf{A})^{-1}. \tag{1.8}$$

Normally, we choose $\mathbf{A}$ evaluated at the time-averaged solution for statistically stationary flows. By performing SVD on the resolvent operator, one can gain insights on the most amplified inputs, and corresponding outputs.

As the last but perhaps most important example of operator-based techniques, the *Koopman operator* (*Mezić*, 2013) is an evolution operator on the infinite dimensional space of observables of non-linear dynamical systems. An introduction to the Koopman operator is provided in the following section.

1.4 Introduction to Koopman Operators

1.4.1 Mathematical Formulation

Consider an autonomous dynamical system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \tag{1.9}$$

where $\mathbf{x} \in \mathcal{M} \subset \mathbb{R}^n$, $\mathcal{M}$ is a smooth manifold in the state space, $\mathbf{F} : \mathcal{M} \mapsto T\mathcal{M}$ is a vector-valued smooth function and $T\mathcal{M}$ is the tangent bundle, i.e., $\forall p \in \mathcal{M}, \mathbf{F}(p) \in T_p\mathcal{M}$. Instead of a geometric viewpoint (*Guckenheimer and Holmes*, 2013; *Hirsch et al.*, 2012), Koopman (*Koopman*, 1931) offered an operator-theoretic perspective by describing the "dynamics of observables" on the measure space $(\mathcal{M}, \Sigma, \mu)$ via the Koopman operator $\mathcal{K}_t : \mathcal{F} \mapsto \mathcal{F}$, such that for an observable on the manifold, $\forall f \in \mathcal{F}$, $t \in \mathbb{R}_{\geqslant 0}$, $f : \mathcal{M} \mapsto \mathbb{C}$,

$$\mathcal{K}_t f \triangleq f \circ S_t, \tag{1.10}$$

where $S_t(\mathbf{x}_0) \triangleq S(\mathbf{x}_0, t)$ is the flow map that takes the initial condition x_0 and advances it by time t by solving the initial value problem for the aforementioned nonlinear dynamics, with $\mathcal{F} = L^2(\mathcal{M}, \mu)$, where μ is some positive measure on $(\mathcal{M}, \Sigma)$. Therefore, it can be also written as

$$\mathcal{K}_t f(\mathbf{x}_0) \triangleq f(\mathbf{x}(t)), \tag{1.11}$$

where $\mathbf{x}(t)$ is the solution of initial value problem of eq. (1.9) given $\mathbf{x}(0) = \mathbf{x}_0$.

The semigroup of Koopman operators $\{\mathcal{K}_t\}_{t \in \mathbb{R}^+}$ is generated by the so-called *Koopman generator* $\mathcal{K} : \mathcal{D}(\mathcal{K}) \mapsto \mathcal{F}$, $\mathcal{K}f \triangleq \lim_{t \to 0}(\mathcal{K}_t f - f)/t$ where $\mathcal{D}(\mathcal{K})$ is the domain in which the aforementioned limit is well-defined and $\overline{\mathcal{D}(\mathcal{K})} = \mathcal{F}$. This operator is *linear*, but defined on $\mathcal{F}$, which makes it inherently *infinite-dimensional*. Physically, $\mathcal{K}_t$ governs the temporal evolution of any observable in $\mathcal{F}$. Specifically, for measure-

preserving systems, e.g., Hamiltonian system or the dynamics on an attractor of the Navier–Stokes equations, as displayed in eq. (1.12) one can guarantee a *spectral decomposition* of any observable $f \in \mathcal{F}$,

$$\mathcal{K}_t f = \underbrace{f^*}_{\text{time-average}} + \underbrace{\sum\nolimits_j e^{\lambda_j t}\mathcal{P}_j f}_{\text{point spectra}} + \underbrace{\int_0^1 e^{i2\pi\theta t} d\mathcal{E}(\theta) f}_{\text{continuous spectra}}, \tag{1.12}$$

where $\lambda_j = i2\pi\omega_j$ is j-th Koopman eigenvalue, and $\mathcal{E}$ is a complex, continuous, operator-valued spectral measure on $\mathcal{F}$.

Similar to the triple decomposition in fluid dynamics (*Hussain and Reynolds*, 1970), such spectral decomposition in eq. (1.12) is a sum of three essentials (*Mezić*, 2005): temporal averaging of the observable, the contribution from the point spectrum, which is *almost periodic* in time, and that from the continuous spectrum, which is *chaotic* (*Koopman and Neumann*, 1932). It should be noted that *pointwise spectra approximation*, i.e., considering the first two terms only, can be quite usually useful already[1] by neglecting the continuous spectra in eq. (1.12). For a more comprehensive discussion, readers are referred to the excellent review by *Budišić et al.* (2012). It should be stressed that for measure-preserving systems, the Koopman operator is not only well-defined on $L^2(\mathcal{M}, \mu)$ but also can be shown to be unitary, which ensures properties such as simple eigenvalues and the existence of the aforementioned spectral resolution (*Budišić et al.*, 2012). Throughout this dissertation, we employ the simple but practical assumption (*Korda and Mezić*, 2018a) $\mathcal{F} = L^2(\mathcal{M}, \mu)$, where μ is some positive measure with support equal to $\mathcal{M}$. Note that this implies that the Koopman operator is well-defined on $\mathcal{F}$.

[1]Indeed, the continuous spectrum is empty for most types of attractors, e.g., fixed points, limit cycle, quasi-periodic tori (*Mauroy and Mezić*, 2016).

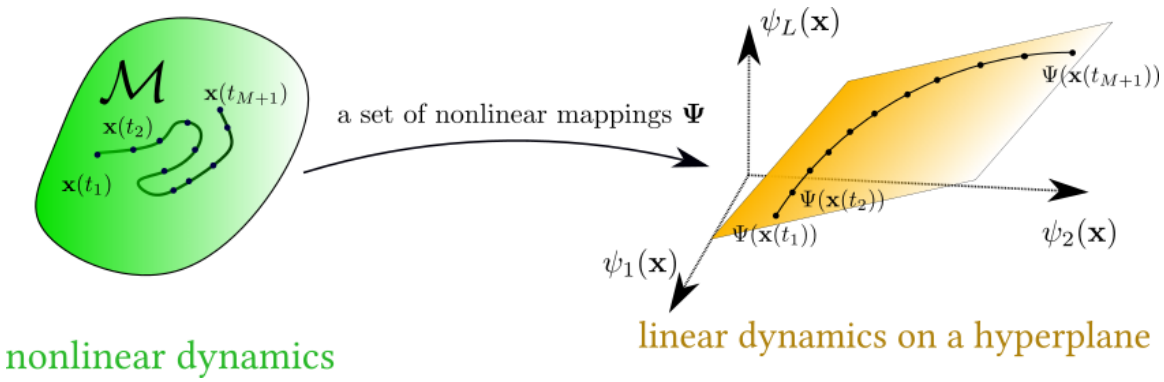


Figure 1.3: Schematic of finding a Koopman-invariant subspace.

1.4.2 Koopman Invariant Subspace & Koopman Analysis

Despite its appealing properties, the Koopman decomposition cannot be directly pursued described above for practical large-scale scientific applications. To accommodate practical computation of the Koopman operator, we adopt the *pointwise spectra assumption*: we assume *only a finite dimensional space invariant to $\mathcal{K}_t$ is of interest.* This excludes the possibility of dealing with a chaotic system since it is impossible for a finite dimensional linear system to be topologically conjugate to a chaotic system (*Budišić et al.*, 2012).[2]

Then, as shown in fig. 1.3, the task of finding Koopman operator becomes equivalent to find a special D dimensional subspace $\mathcal{F}_D$ in the observables space $\mathcal{F}$ that is *invariant* to Koopman operator. That is to say, for any ϕ in the $\mathcal{F}_D$, for any $t \in \mathbb{R}^+$, we have $\mathcal{K}_t\phi$ also in the $\mathcal{F}_D$. Further, as shown in eq. (1.13), we refer *Koopman analysis* of a dynamical system in eq. (1.9) to searching for the Koopman eigenfunctions $\psi_1, \ldots, \psi_D$ that span a non-trivial $\mathcal{F}_D$ that contains the identity mapping $f = \mathbf{x}$ and its associated eigenvalues $\lambda_1, \ldots, \lambda_D$[3].

[2]However, one could assign a state-dependent Koopman operator to work around this limitation (*Lusch et al.*, 2018a). Indeed, one can establish a duality between the measure-preserving dynamics on the attractor with a stationary stochastic process, which reflects a close relationship between the estimation of a *continuous spectrum* from the data, and the extraction of the power spectral density of stochastic signals (*Arbabi and Mezić*, 2017).

[3]While we assume that the eigenvalues are *simple* in this dissertation, it is possible to extend to the generalized case (*Budišić et al.*, 2012).

$$\underbrace{\mathbf{x}(t)}_{\text{non-linear dynamics}} = \sum_{j=1}^{D} \underbrace{\psi_j(\mathbf{x}) e^{\lambda_j t} \mathbf{b}_j}_{\text{single mode of linear dynamics}} , \tag{1.13}$$

where we call $\mathbf{b}_1, \ldots, \mathbf{b}_D$, the projections coefficients of $\mathbf{x}$ onto the eigenfunctions as *Koopman modes*.

Such non-trivial $\mathcal{F}_D$ would correspond to a linear system that is topologically conjugate to the nonlinear dynamical system. This concept is particularly useful in high-dimensional system such as fluid dynamics. Moreover, such decomposition essentially generalizes the Hartman-Grobman theorem to the entire basin of attraction of the equilibrium point or periodic orbit (*Lan and Mezić*, 2013). From the viewpoint of understanding the behavior of the dynamical system in state space, the level sets of Koopman eigenfunctions form the invariant partition in the state space (*Budišić and Mezić*, 2012). Further, the ergodic partition can be analyzed with Koopman eigenfunctions (*Mezić and Wiggins*, 1999).

1.4.3 Implications of Koopman Operators

1. **Physical insight**. Complex unsteady flow phenomena such as turbulence (*Pope*, 2001), flow instability (*Drazin and Reid*, 2004; *Lietz et al.*, 2017), and fluid-structure interactions (*Dowell and Hall*, 2001) are prevalent in many physical systems. To analyze and understand such phenomena, it is useful to extract coherent modes associated with important dynamical mechanisms and track their evolution. Koopman operator theory (*Budišić et al.*, 2012) offers an elegant framework to reduce spatio-temporal fields associated with non-linear dynamics as a linear combination of time evolving modes ordered by isolated frequencies and growth rates (*Rowley et al.*, 2009). For example, DMD which is the simplest approximation of the Koopman operator, has been widely applied in transitional flows (*Sayadi et al.*, 2012, 2014), acoustics (*Song et al.*, 2013),

combustion instability (*Arnold-Medabalimi et al.*, 2020; *Huang et al.*, 2016), etc. From the viewpoint of understanding the behavior of the dynamical system in state space, the level sets of Koopman eigenfunctions form the invariant partition in the state space (*Budišić and Mezić*, 2012), which can help study mixing. Further, the ergodic partition can be analyzed with Koopman eigenfunctions (*Mezić and Wiggins*, 1999).

2. **ROMs for prediction, control & state estimation**. The Koopman operator naturally induces a low-dimensional model that can enable predictive modeling and the possibility of real-time control. It is notable that controllers based on linear optimal control theory can be implemented efficiently (*Deem et al.*, 2020). On one hand, in terms of reduced order modeling for prediction, we are interested in learning the reduced dynamics from data. Intrusive ROMs (*Huang et al.*, 2018; *Carlberg et al.*, 2013; *Huang et al.*, 2020a) involve stability and efficiency issues when applied to stiff and convection-dominated chaotic problems (e.g., combustion). Although non-intrusive (NI) ROMs (*Xu and Duraisamy*, 2019) show can be more efficient, there are typically no solid guarantees for the long time prediction to be accurate and stable. In contrast, Koopman-based NI-ROMs have two major advantages over standard NI-ROM. First, generally speaking, the off-line computational complexity of Koopman-based NI-ROM is lower than standard NI ROM in that one only needs to evaluate a matrix exponential for the prediction at an arbitrary future time step. This can be particularly attractive for real-time forecasting applications. Meanwhile, standard NI-ROMs have to integrate non-linear latent dynamics which is much more computational costly than evaluating a matrix exponential. Second, one can elegantly enforce stability within the framework of Koopman operator (*Pan and Duraisamy*, 2020c) while such integration of non-linear latent dynamics from standard NI-ROMs could lead to uncontrolled instability.

It has to be mentioned, however, that a linear dynamics may demand a much higher dimensional latent space than would be required if non-linear dynamics is pursued.

Koopman-based system identification models are particularly desirable in data-driven control due to the potential to leverage linear optimal control theory (*Abraham et al.*, 2017; *Mamakoukas et al.*, 2019; *Kaiser et al.*, 2017), linear model predictive control (MPC) (*Korda and Mezić*, 2018b) and state estimation (*Surana and Banaszuk*, 2016). This is particularly attractive for identifying non-linear systems where the full dynamics is either too high dimensional to handle (e.g., flow control (*Arbabi et al.*, 2018b)), contains important missing physics and uncertainty (e.g., frictions, contact dynamics) or just simply difficult to obtain (e.g., soft robotics (*Bruder et al.*, 2019)). For example, the main idea of Koopman-MPC (*Korda and Mezić*, 2016) is to consider the *approximation* of the original non-linear dynamical system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}), \tag{1.14}$$

with a linear time-variant input-output system in the Koopman lifted space,

$$\dot{\mathbf{\Psi}} = \mathbf{A}\mathbf{\Psi} + \mathbf{B}\mathbf{u}, \quad \mathbf{x} = \mathbf{C}\mathbf{\Psi}(\mathbf{x}). \tag{1.15}$$

It is thus natural to leverage the infinite-horizon linear quadratic regulator (LQR) in the form

$$\mathbf{u}^*(t) = \arg\min_{\mathbf{u}(t)} J(\mathbf{u}, \mathbf{x}_0) = \arg\min_{\mathbf{u}(t)} \int_0^\infty \left(\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right) dt, \tag{1.16}$$

$$= \arg\min_{\mathbf{u}(t)} \int_0^\infty \left(\mathbf{\Psi}^\top \mathbf{Q}_{\mathbf{\Psi}} \mathbf{\Psi} + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right), \tag{1.17}$$

where the closed form solution induces a *non-linear* feedback controller:

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}\boldsymbol{\Psi}(\mathbf{x}). \tag{1.18}$$

and $\mathbf{P}$ can be solved via the following algebraic Ricatti equation just for one time,

$$\mathbf{Q}_{\boldsymbol{\Psi}} + \mathbf{P}\mathbf{A} + \mathbf{A}^\top \mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} = \mathbf{0}. \tag{1.19}$$

3. **Operator-informed deep learning architectures**. One of the most notorious problems in training deep recurrent neural networks (RNN) is the issue of vanishing gradient. This is a result of the contracting behavior when information is propagated in each layer, and hampers learning from long sequences. One of the popular remedies is "Long short-term memory(LSTM)" (*Hochreiter and Schmidhuber*, 1997). Even though applying Koopman operator theory with a finite-dimensional linear representation of fully chaotic systems is less ideal compared to the non-chaotic counterparts, several studies from different groups recently reported by *Otto and Rowley* (2019b); *Morton et al.* (2019); *Eivazi et al.* (2020) have all demonstrated favorable performance of the Koopman operator in modeling high-dimensional chaotic systems over LSTM. This benefit is a consequence of the *linear embedding* which makes the information propagation much easier in the recurrent structure. This can be generalized to many other deep learning applications (e.g., video predictions) by equipping CNN with Koopman operator (*Leask and McDonell*, 2020; *Gin et al.*, 2019).

1.5 Overview of the state-of-the art in the Approximation of Koopman Operators

In this section, we provide a high-level overview of the state-of-the-art in the approximation of Koopman operators. A more detailed presentation is provided in Chapter II. Since we adopt the *pointwise spectral assumption*, i.e., we neglect the continuous spectrum, our goal is to seek a finite number of Koopman eigenfunctions and corresponding eigenvalues. Existing methods for the discovery of such eigenfunctions and eigenvalues can be broadly classified into two kinds:

- *Analytic approximations*: These techniques directly manipulate the PDE to find the Koopman decomposition. For example, *Bagheri* (2013) considered a weakly non-linear analysis with 1D Stuart-Landau equation to obtain Koopman eigenvalues and eigenfunctions for 2D cylinder flow; *Page and Kerswell* (2018); *Nathan Kutz et al.* (2018) leveraged Cole-Hopf transformation to find Koopman decompositions for the 1D Burgers equation.

- *Data-enabled/data-driven approximations*: This class of techniques use the state or observable data and/or the governing equations to search for Koopman eigenfunctions in an appropriate function space.

Since the analytical approach is restricted to a few simple problems, we focus our attention on data-driven and data-enabled techniques to approximate Koopman operators. The most straightforward idea (*Williams et al.*, 2015) is to manually construct a finite number of so-called *dictionary* functions of the state $\mathbf{x}$, and then assume that the space spanned by these functions is invariant to the Koopman operator. Then the realization of Koopman operator becomes a finite-dimensional square matrix. Without loss of generality, we consider a trajectory of the original non-linear dynamical system: $\mathbf{x}(\Delta t), \ldots, \mathbf{x}(M\Delta t)$. The simplest way to obtain such a square matrix is to fit a linear mapping between dictionary functions at time t and those

at time $t + \Delta t$. Then one can simply perform an eigendecomposition to obtain the corresponding approximation of Koopman eigenvalues and eigenfunctions. Based on *how one constructs such dictionary functions*, there are three different categories of approximation methods:

1. **DMD** (*Rowley et al.*, 2009), **Extended DMD** (*Williams et al.*, 2015), **Kernel DMD** (*Williams et al.*, 2014): These methods use fixed dictionary functions, either from linear features, polynomial features, or implicit features via the kernel trick. The latter two methods yield *non-linear* approximations to the Koopman operator, whereas while the former is a *linear* approximation (*Rowley et al.*, 2009). Such dictionary-based methods (e.g., EDMD) enjoy rigorous theoretical justification as a L^2 projection of the Koopman operator in a measure-theoretic sense (*Korda and Mezić*, 2018a). Moreover, when an appropriate space is chosen, these techniques converge in the strong operator topology to the Koopman operator with infinite number of dictionary functions. In practice, non-linear approximations of Koopman operator have been shown to improve the accuracy of approximation compared to linear counterpart (DMD) (*Pan et al.*, 2020). However, due to its simplicity, DMD is much more popular in the fluid dynamics community (*Schmid*, 2010; *Schmid et al.*, 2011).

2. **Time delay DMD** (*Brunton et al.*, 2017): Time-delay embedding, also known as delay-coordinate embedding, refers to the inclusion of history information in dynamical system models. This idea has been employed in a wide variety of contexts especially in Koopman operators (*Arbabi and Mezic*, 2017; *Arbabi and Mezić*, 2017; *Kamb et al.*, 2018; *Brunton et al.*, 2017). The use of delays to construct a "rich" feature space for geometrical reconstruction of non-linear dynamical systems is justified by the Takens embedding theorem (*Takens*, 1981) which states that by using a *delay-coordinate map*, one can construct a diffeomorphic shadow manifold from univariate observations of the original system in

the generic sense. Extensions are possible in a measure-theoretic sense (*Sauer et al.*, 1991), filtered memory (*Sauer et al.*, 1991), deterministic/stochastic forcing (*Stark et al.*, 2003b,a), and multivariate embeddings (*Deyle and Sugihara*, 2011). In the context of the Koopman operator, instead of considering manually built dictionary functions in the previous category, one can consider using the previous states of the system as augmented observables. Interestingly, this can be equivalently viewed as taking a series of time-delayed reverse flow maps as fixed dictionary functions with the current state as the initial condition. Such a dictionary function is, however, implicitly defined by the dynamics of the autonomous system and most of the time one cannot find an explicit expression[4]. Therefore, one has to either collect experimental measurements for the initialization or run numerical simulations to obtain enough history data to start the model for prediction purposes.

3. **Neural Networks** (*Lusch et al.*, 2018b): Neural networks drives many state-of-the-art deep learning applications (*Krizhevsky et al.*, 2017; *Ren et al.*, 2015). They provides a special compositional parameterization that can approximate arbitrary functions (see appendix B.1.3). Neural networks can be used to construct observables, following which one can penalize the deviation from linear dynamics (*Otto and Rowley*, 2019b; *Yeung et al.*, 2017). The benefits of this approach compared to previous methods is the expressiveness and the potential to be adaptive to the data while still enjoying explicit transformations. However, one need to pre-determine the dimension of the Koopman invariant subspace and such training process suffers from issues of non-convex optimization.

As a side note, other algorithms for the approximation of the Koopman operator (but not included in this thesis) include generalized Laplace analysis (GLA) (*Budišić et al.*, 2012; *Mauroy and Mezić*, 2012), and the Ulam Galerkin method (*Froyland*

[4]One needs to have an explicit expression for the flow map.

et al., 2014).

1.6 Motivating Questions

From the standpoint of the approximation of Koopman operators, the following fundamental questions serve as the motivation for this work:

1. How best can one leverage time delays in DMD?: What is the minimal number of time delays required to develop accurate approximations? From a numerical conditioning perspective, how do time delays affect model performance?

2. Since we assume that the space spanned by *all* dictionary functions is Koopman invariant, how can we choose a *sparse* and *robust* Koopman invariant subspace from data-driven nonlinear approximations for Koopman operators, e.g., Extended DMD, Kernel DMD?

3. Beyond the black-box nature of deep neural networks, how can we use physics information and how can we guarantee the stability of the Koopman approximation? Further, how can we address uncertainties arising from finite data and noise? As a consequence, how can we quantify the confidence in Koopman operator-based predictions?

The goal of this dissertation is to address the above questions and advance the theoretical & numerical understanding of Koopman operator approximations.

1.7 Contributions

As summarized in fig. 1.4, the key contributions of this dissertation are:

1. Theoretical explanation of the role of time delays in data-driven Koopman approximation. This includes quantitative details on the number of time delays

required in linear embedding of non-linear dynamics on an attractor, and the impact of time delays on numerical conditioning.

2. Multi-task learning and error analysis to choose an optimal basis which dramatically reduces the dimension of the Koopman invariant subspace.

3. Theoretical analysis of the connection between a empirical criterion, sparse-promoting DMD and the proposed multi-task learning framework from non-convex optimization.

4. Successful applications on several transient fluid flows demonstrate the power of extracting correct coherent structures without the need to separate training data for the strongly non-linear dynamics (*Taira et al.*, 2020).

5. A complete measure-theoretic formulation of data-driven approximation of Koopman operators with deep neural networks.

6. Devised and proved a technique to enforce non-linear stability in Koopman operators with no degradation on expressiveness.

7. Developed a hierarchical Bayesian deep learning framework to address the uncertainty in the approximation of Koopman operators.

The above contributions are also presented in the following publications:

1. **Pan, S.**, and Duraisamy, K. (2020) Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM Journal on Applied Dynamical Systems*, 19(1), 480-509

2. **Pan, S.**, and Duraisamy, K. (2020). On the structure of time-delay embedding in linear models of non-linear dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30 (7), 073135

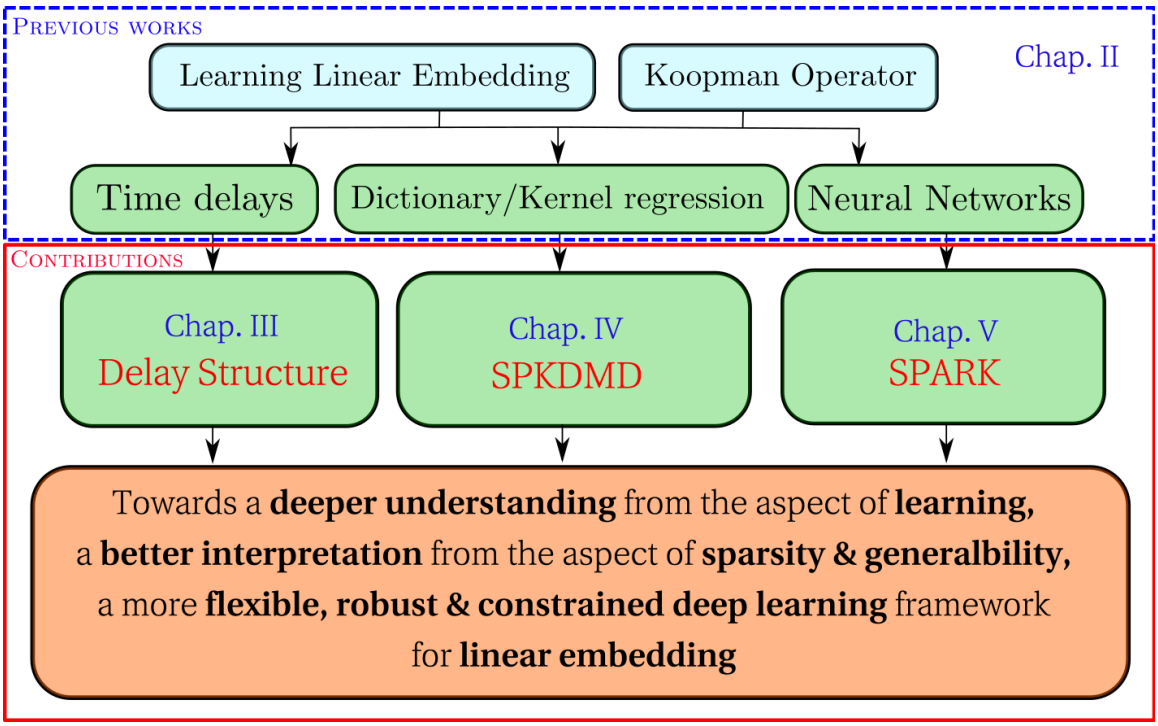


Figure 1.4: Sketch of contributions in this dissertation.

3. **Pan, S.**, Arnold-Medabalimi, N., and Duraisamy, K. (2020). Sparsity-promoting algorithms for the discovery of informative Koopman invariant subspaces. arXiv preprint arXiv:2002.10637. (conditionally accepted *Journal of Fluid Mechanics*)

While not explicitly included in the thesis, the following papers - written over the duration of the PhD - served to further refine numerical treatments and overarching questions in this dissertation:

1. **Pan, S.**, and Duraisamy, K. (2018). Data-driven discovery of closure models. *SIAM Journal on Applied Dynamical Systems*, 17(4), 2381-2413.

2. **Pan, S.**, and Duraisamy, K. (2018). Long-time predictive modeling of nonlinear dynamical systems using neural networks. *Complexity*, 2018.

3. Singh, A. P., Duraisamy, K., and **Pan, S**. (2017). Augmentation of turbulence models using field inversion and machine learning. In *55th AIAA Aerospace Sciences Meeting* (p. 0993).

4. Sun, L., Gao, H., **Pan, S.**, and Wang, J. X. (2020). Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361, 112732.

5. Ji, W., Qiu, W., Shi, Z., **Pan, S.**, and Deng, S. (2020). Stiff-PINN: Physics-Informed Neural Network for Stiff Chemical Kinetics. *arXiv preprint* arXiv:2011.04520.

6. Singh, A. P., **Pan, S.** and Duraisamy, K., (2017). Characterizing and improving predictive accuracy in shock-turbulent boundary layer interactions using data-driven models. In *55th AIAA Aerospace Sciences Meeting* (p. 0314).

7. Gao, Q., Li, Q., **Pan, S.**, Wang, H., Wei, R., and Wang, J. (2019). Particle reconstruction of volumetric particle image velocimetry with strategy of machine learning. *arXiv preprint* arXiv:1909.07815.

8. Bhatnagar, S., Afshar, Y., **Pan, S.**, Duraisamy, K., and Kaushik, S. (2019). Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2), 525-545.

1.8 Outline

The structure of the dissertation is as follows: Chapter II presents a review of the state-of-the-art in data-driven approximation of Koopman operators and remaining challenges. In the next following three chapters, the major contributions of this thesis are presented. Chapter III presents the theory on the role of the number of delays in time delay DMD restricted to a periodic attractor with several validations including chaotic turbulent flows. Chapter IV addresses the issue extracting compact Koopman invariant subspaces from traditional methods (EDMD/KDMD) using a multi-task learning framework with error analysis. Several applications on transient non-linear dynamical systems are also provided. Chapter V presents a Bayesian deep learning

framework with stability guarantees to extract Koopman operators with and without the use of data. Finally, conclusions and perspectives are provided in Chapter VI.

CHAPTER II

Data-Driven Approximation of Koopman Operators

> Truth ... is much too complicated to allow anything but approximations.
>
> *John von Neumann (1947)*

2.1 Introduction

In section 1.4.2, we showed that the goal of extracting Koopman operators for non-linear dynamical systems under the assumption of *pointwise spectra* is equivalent to finding a finite set of Koopman eigenfunctions $\psi_1, \ldots, \psi_D$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_D$. Different methods of approximating the Koopman operator were briefly introduced in section 1.5. In this chapter, we will provide a more detailed description of the state-of-the-art.

We also develop the mathematical setting necessary for the advances pursued in this thesis.

The three categories of data-driven methods for the approximation the Koopman eigenfunctions and eigenvalues are:

- EDMD/KDMD. This is perhaps the most common approach for approximating

the Koopman operator. The main idea behind EDMD (*Williams et al.*, 2015) and KDMD (*Williams et al.*, 2014) is to assume that the Koopman invariant subspace $\mathcal{F}_D$ is spanned by a finite dictionary of functions given by the user (e.g., monomials or implicit functions defined by the kernel (*Mercer*, 1909a)), and then minimize the empirical risk of the residual that comes from the imperfect Koopman invariant subspace. Note that EDMD is essentially a least-squares regression of the (linear) action of the Koopman operator for the features in the dictionary. Such samples are drawn from some measure μ with fixed features, which can be proved as a $L^2(\mathcal{M}, \mu)$ *orthogonal projection* in the limit of large independent identically distributed data (*Klus et al.*, 2015; *Williams et al.*, 2015). Indeed, by framing the data-derived Koopman operator in the Hilbert space endowed with a proper measure, one can prove optimality in the asymptotic sense with an infinite amount of data and dictionary functions for EDMD (*Korda and Mezić*, 2018a). For Hankel-DMD, convergence to the Koopman eigenfunctions and eigenvalues can be guaranteed for an ergodic system with an infinite amount of data (*Arbabi and Mezic*, 2017). Unfortunately, the original EDMD (*Williams et al.*, 2015) is seldom used for data-driven Koopman analysis in high-dimensional systems such as fluid flows due to its poor scaling with the dimension of the system state. However, one can leverage *random Fourier features* (*Rahimi and Recht*, 2008) to construct a scalable version of EDMD (*DeGennaro and Urban*, 2019) for high-dimensional systems.

- Time Delay DMD. Following the similar spirit of EDMD, time delay DMD takes current and past L system states on the trajectory to construct an observable space that approximates the Koopman invariant subspace. We formalize it in discrete-time, which can be viewed as taking a sequence of flow maps with different time intervals for constructing the observable space to build an orthogonal projection of the Koopman operator. As a side note, it is possible to formalize

in continuous-time with the inclusion of Lie derivative.

- Deep learning-based DMD (DL-DMD) in *differential* form and *recurrent* form. *Differential form* refers to enforcing the time derivative of Koopman eigenfunctions within the subspace spanned by the eigenfunctions. *Recurrent form* refers to enforcing the recurrent prediction from the Koopman-based model to match the discretely sampled trajectory data. Further, we formalize the DL-DMD with statistical learning concepts in a probabilistic sense.

2.2 Extended Dynamic Mode Decomposition

2.2.1 Discrete-time Formulation

For simplicity, consider M sequential snapshots sampled uniformly in time with Δt on a trajectory, $\{\mathbf{x}_i\}_{i=1}^{M}$. The EDMD algorithm (*Williams et al.*, 2015) assumes a dictionary of L linearly independent functions $i = 1, \ldots, L$, $\psi_i(\mathbf{x}) : \mathcal{M} \mapsto \mathbb{C}$ that approximately spans a Koopman-invariant subspace $\mathcal{F}_L$

$$\mathcal{F}_L = \text{span}\{\psi_1(\mathbf{x}), \ldots, \psi_L(\mathbf{x})\}. \tag{2.1}$$

Thus we can write for any $g \in \mathcal{F}_L$ as $g(\mathbf{x}) = \mathbf{\Psi}(\mathbf{x})\mathbf{a}$ with $\mathbf{a} \in \mathbb{C}^L$, where the feature vector $\mathbf{\Psi}(\mathbf{x})$ is

$$\mathbf{\Psi}(\mathbf{x}) = \begin{bmatrix} \psi_1(\mathbf{x}) & \ldots & \psi_L(\mathbf{x}) \end{bmatrix}. \tag{2.2}$$

Consider a set of $L^{'}$ observables as $\{g_l\}_{l=1}^{L^{'}} = \{\mathbf{\Psi}\mathbf{a}_l\}_{l=1}^{L^{'}}$, where $\mathbf{a}_l \in \mathbb{C}^L$ is arbitrary. After the discrete-time Koopman operator $\mathcal{K}_{\Delta t}$ is applied on each g_l, given data $\{\mathbf{x}_i\}_{i=1}^{M}$, we have the following for $l = 1, \ldots, L'$, and $i = 1, \ldots, M-1$,

$$\mathcal{K}_{\Delta t} g_l(\mathbf{x}_i) = g_l(\mathbf{x}_{i+1}) = \mathbf{\Psi}(\mathbf{x}_{i+1})\mathbf{a}_l = \mathbf{\Psi}(\mathbf{x}_i)\mathbf{K}\mathbf{a}_l + r_{i,l}, \tag{2.3}$$

where $r_{i,l}$ is simply the residual for the l-th function on the i-th data point. The standard EDMD algorithm seeks a constant matrix $\mathbf{K} \in \mathbb{C}^{L \times L}$ that governs the linear dynamics in the observable space such that the sum of the square of the residuals $r_{i,l}$ from eq. (2.3) over all samples and functions,

$$J(\mathbf{K}, \{\mathbf{a}_l\}_{l=1}^{L'}) = \sum_{l=1}^{L'} \sum_{m=1}^{M-1} |(\mathbf{\Psi}(\mathbf{x}_{m+1}) - \mathbf{\Psi}(\mathbf{x}_m)\mathbf{K})\mathbf{a}_l|^2 = \|(\mathbf{\Psi}_{\mathbf{x}'} - \mathbf{\Psi}_{\mathbf{x}}\mathbf{K})\mathbf{A}'\|_F^2, \quad (2.4)$$

is minimized over $\{\mathbf{x}_i\}_{i=1}^{M+1}$. In the above equation,

$$\mathbf{\Psi}_{\mathbf{x}} = \begin{bmatrix} \psi_1(\mathbf{x}_1) & \dots & \psi_L(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ \psi_1(\mathbf{x}_{M-1}) & \dots & \psi_L(\mathbf{x}_{M-1}) \end{bmatrix}, \mathbf{\Psi}_{\mathbf{x}'} = \begin{bmatrix} \psi_1(\mathbf{x}_2) & \dots & \psi_L(\mathbf{x}_2) \\ \vdots & \vdots & \vdots \\ \psi_1(\mathbf{x}_M) & \dots & \psi_L(\mathbf{x}_M) \end{bmatrix} \quad (2.5)$$

$$\mathbf{A}' = \begin{bmatrix} \mathbf{a}_1 & \dots & \mathbf{a}_{L'} \end{bmatrix}. \quad (2.6)$$

Considering $\partial J / \partial \mathbf{K} = \mathbf{0}$, one obtains $\mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}} \mathbf{\Psi}_{\mathbf{x}'} \mathbf{A}' \mathbf{A}'^{\mathrm{H}} = \mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}} \mathbf{\Psi}_{\mathbf{x}} \mathbf{K} \mathbf{A}' \mathbf{A}'^{\mathrm{H}}$. Thus, the corresponding minimizer $\mathbf{K}_{\mathrm{opt}}$ is,

$$\mathbf{K}_{\mathrm{opt}} = \mathbf{G}^{+} \mathbf{A} (\mathbf{A}' \mathbf{A}'^{\mathrm{H}}) (\mathbf{A}' \mathbf{A}'^{\mathrm{H}})^{+}, \quad (2.7)$$

where + denotes the pseudoinverse and

$$\mathbf{G} = \sum_{m=1}^{M-1} \mathbf{\Psi}(\mathbf{x}_m)^{\mathrm{H}} \mathbf{\Psi}(\mathbf{x}_m) = \mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}} \mathbf{\Psi}_{\mathbf{x}}, \quad (2.8)$$

$$\mathbf{A} = \sum_{m=1}^{M-1} \mathbf{\Psi}(\mathbf{x}_m)^{\mathrm{H}} \mathbf{\Psi}(\mathbf{x}_{m+1}) = \mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}} \mathbf{\Psi}_{\mathbf{x}'}, \quad (2.9)$$

where H denotes conjugate transpose. Note that when the set of observable fully span $\mathcal{F}_L$, i.e., $\mathbf{A}'$ is full rank, $(\mathbf{A}' \mathbf{A}'^{\mathrm{H}})(\mathbf{A}' \mathbf{A}'^{\mathrm{H}})^{+}$ reduces to identity. Then we have

the more familiar $\mathbf{K}_{\text{EDMD}}$ as,

$$\mathbf{K}_{\text{EDMD}} = \mathbf{G}^{+}\mathbf{A}. \tag{2.10}$$

which is independent of *the choice of* $\{\mathbf{a}_l\}_{l=1}^{L'}$.

Assuming that all eigenvalues of $\mathbf{K}_{\text{EDMD}}$ are simple[1], for $i = 1, \ldots, L$, the corresponding Koopman eigenfunctions φ_i associated with Koopman eigenvalues λ_i are

$$\varphi_i(\mathbf{x}) = \mathbf{\Psi}(\mathbf{x})\mathbf{v}_i, \tag{2.11}$$

where $\mathbf{K}_{\text{EDMD}}\mathbf{v}_i = \lambda_i \mathbf{v}_i$. Finally, the Koopman modes of a vector-valued Q dimensional observable function $\mathbf{g} : \mathcal{M} \mapsto \mathbb{C}^Q$, are the vectors of weights $\{\mathbf{b}_i\}_{i=1}^{L}$ assoicated with the expansion in terms of eigenfunctions $\{\varphi_i\}_{i=1}^{L}$ as,

$$\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{L} \varphi_i(\mathbf{x})\mathbf{b}_i, \tag{2.12}$$

where $\mathbf{b}_i$ is often numerically determined in practice by ordinary least squares,

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_L \end{bmatrix} = (\mathbf{\Psi}_{\mathbf{x}}\mathbf{V})^{+} \begin{bmatrix} \mathbf{g}(\mathbf{x}_1) \\ \vdots \\ \mathbf{g}(\mathbf{x}_M) \end{bmatrix}, \tag{2.13}$$

with $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \ldots & \mathbf{v}_L \end{bmatrix}$.

2.2.2 Continuous-time Formulation

Consider M data snapshots of the dynamical system with state $\mathbf{x}$ sampled over $\mathcal{M}$ as $\{\mathbf{x}_i, \dot{\mathbf{x}}_i\}_{i=1}^{M}$ where $\dot{\mathbf{x}}_i = \mathbf{F}(\mathbf{x}_i)$. Recall the generator of the semigroup of Koopman operators $\mathcal{K} : \mathcal{D}(\mathcal{K}) \mapsto \mathcal{F}, \mathcal{K} = \lim_{t\to 0}(\mathcal{K}_t f - f)/t$ where $\mathcal{D}(\mathcal{K})$ is the domain in which the aforementioned limit is well-defined and the closure of $\mathcal{D}(\mathcal{K})$ is $\mathcal{F}$. One can have

[1]This is an immediate consequence for ergodic system (*Parry*, 2004).

the evolution of any observable $\mathbf{g} = \mathbf{\Psi a} \in \mathcal{F}_L$ as,

$$\mathcal{K}\mathbf{g} = \dot{\mathbf{g}} = \mathbf{F} \cdot \nabla_{\mathbf{x}} \mathbf{\Psi a} = \mathbf{\Psi K a} + r, \tag{2.14}$$

where r is the residual. Similarly, one can find a $\mathbf{K}$ that minimizes the sum of the square of residual r minimized solution,

$$\mathbf{K}_{\text{EDMD}} = \mathbf{G}^{+}\mathbf{A}, \tag{2.15}$$

where

$$\mathbf{G} = \sum_{m=1}^{M} \mathbf{\Psi}(\mathbf{x}_m)^{\text{H}} \mathbf{\Psi}(\mathbf{x}_m), \tag{2.16}$$

$$\mathbf{A} = \sum_{m=1}^{M} \mathbf{\Psi}(\mathbf{x}_m)^{\text{H}} (\mathbf{F} \cdot \nabla_{\mathbf{x}}) \mathbf{\Psi}(\mathbf{x}_m) = \sum_{m=1}^{M} \mathbf{\Psi}(\mathbf{x}_m)^{\text{H}} (\dot{\mathbf{x}}_m \cdot \nabla_{\mathbf{x}}) \mathbf{\Psi}(\mathbf{x}_m). \tag{2.17}$$

Consider eigenvalues $\{\mu_i\}_{i=1}^{L}$ and eigenvectors $\{\mathbf{v}_i\}_{i=1}^{L}$ of $\mathbf{K}_{\text{EDMD}}$. Koopman eigenfunctions are in the same form as that in discrete-time formulations while continuous-time Koopman eigenvalues μ_i can be converted to the aforementioned discrete-time sense as $\lambda_i = e^{\mu_i \Delta t}$.

2.3 Kernel Dynamic Mode Decomposition

2.3.1 Discrete-time Formulation

Instead of explicitly expressing a dictionary of candidate functions, one can instead implicitly define a dictionary of candidate functions via the kernel trick, which is essentially the dual form of EDMD (*Williams et al.*, 2014). Note that, from the EDMD formulation in eq. (2.4), any vector in the range of $\mathbf{K}$ orthogonal to the range of $\mathbf{\Psi}_{\mathbf{x}}^{\text{H}}$ is annihilated, and therefore cannot be inferred (*Williams et al.*, 2014). Assuming $\mathbf{\Psi}_{\mathbf{x}}$ is of rank r, we can obtain a full SVD $\mathbf{\Psi}_{\mathbf{x}} = \mathbf{Q\Sigma Z}^{\text{H}}$ and the corresponding

economical-SVD as $\mathbf{Q}_r\mathbf{\Sigma}_r\mathbf{Z}_r^{\mathrm{H}}$. With the aforementioned definitions, we have $\mathbf{K} = \mathbf{Z}_r\hat{\mathbf{K}}\mathbf{Z}_r^{\mathrm{H}}$ without loss (*Otto and Rowley*, 2019b).

Since the multiplication by a unitary matrix preserves the Frobenius norm, we have

$$J(\mathbf{K}, \{\mathbf{a}_l\}_{l=1}^{L'}) = \|(\mathbf{\Psi}_{\mathbf{x}'} - \mathbf{\Psi}_{\mathbf{x}}\mathbf{K})\mathbf{A}'\|_F^2 \tag{2.18}$$

$$= \|(\mathbf{Q}^{\mathrm{H}}\mathbf{\Psi}_{\mathbf{x}'} - \mathbf{Q}^{\mathrm{H}}\mathbf{Q}_r\mathbf{\Sigma}_r\hat{\mathbf{K}}\mathbf{Z}_r^{\mathrm{H}})\mathbf{A}'\|_F^2 \tag{2.19}$$

$$= \|(\mathbf{Q}_r^{\mathrm{H}}\mathbf{\Psi}_{\mathbf{x}'} - \mathbf{\Sigma}_r\hat{\mathbf{K}}\mathbf{Z}_r^{\mathrm{H}})\mathbf{A}'\|_F^2 + \|\mathbf{Q}_{r,\perp}^{\mathrm{H}}\mathbf{\Psi}_{\mathbf{x}'}\mathbf{A}'\|_F^2, \tag{2.20}$$

where $\mathbf{Q}_{r,\perp}^{\mathrm{H}}$ are the last $m - r$ rows of $\mathbf{Q}^{\mathrm{H}}$. By taking derivatives with respect to $\hat{\mathbf{K}}$, one can obtain the minimal-norm minimizer as,

$$\hat{\mathbf{K}}_{\mathrm{opt}} = \mathbf{\Sigma}_r^{+}\mathbf{Q}_r^{\mathrm{H}}\mathbf{\Psi}_{\mathbf{x}'}\mathbf{A}'\mathbf{A}'^{\mathrm{H}}\mathbf{Z}_r(\mathbf{Z}_r^{\mathrm{H}}\mathbf{A}'\mathbf{A}'^{\mathrm{H}}\mathbf{Z}_r)^{+}. \tag{2.21}$$

Notice that, since any column in the span of $\mathbf{A}'$ that is orthogonal to the span of $\mathbf{Z}_r$ will be annihilated by $\mathbf{Z}_r^{\mathrm{H}}$ and thus cannot be utilized to determine $\hat{\mathbf{K}}$, it is reasonable to restrict $\mathbf{a}_l$ within the column space of $\mathbf{Z}_r$. Assuming L' is sufficiently large such that the column space of $\mathbf{A}'$ fully spans $\mathbf{Z}_r$, eq. (2.22) can be proved as follows:

$$\mathbf{A}'\mathbf{A}'^{\mathrm{H}}\mathbf{Z}_r(\mathbf{Z}_r^{\mathrm{H}}\mathbf{A}'\mathbf{A}'^{\mathrm{H}}\mathbf{Z}_r)^{+} = \mathbf{Z}_r. \tag{2.22}$$

Proof. Consider the economy SVD of $\mathbf{A}' = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{H}}$. Since the column space of $\mathbf{A}'$ spans the column space of $\mathbf{Z}_r$, there exists an invertible matrix $\mathbf{P}$ such that $\mathbf{Z}_r\mathbf{P} = \mathbf{U}$. Hence,

$$\mathbf{A}'\mathbf{Z}_r = \mathbf{V}\mathbf{S}\mathbf{U}^{\mathrm{H}}\mathbf{Z}_r = \mathbf{V}\mathbf{S}\mathbf{P}^{\mathrm{H}}\mathbf{Z}_r^{\mathrm{H}}\mathbf{Z}_r = \mathbf{V}\mathbf{S}\mathbf{P}^{\mathrm{H}},$$

and $((\mathbf{A}'\mathbf{Z}_r)^{\mathrm{H}}\mathbf{A}'\mathbf{Z}_r)^{+} = (\mathbf{P}^{-1})^{\mathrm{H}}\mathbf{S}^{-2}\mathbf{P}^{-1}$. Thus

$$\mathbf{A}'\mathbf{A}'^{\mathrm{H}}\mathbf{Z}_r(\mathbf{Z}_r^{\mathrm{H}}\mathbf{A}'\mathbf{A}'^{\mathrm{H}}\mathbf{Z}_r)^{+} = (\mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{H}})(\mathbf{V}\mathbf{S}\mathbf{P}^{\mathrm{H}})(\mathbf{P}^{-1})^{\mathrm{H}}\mathbf{S}^{-2}\mathbf{P}^{-1} = \mathbf{U}\mathbf{P}^{-1} = \mathbf{Z}_r.$$

□

With eq. (2.22), we can rewrite eq. (2.21) as the familiar KDMD formulation,

$$\hat{\mathbf{K}}_{\mathrm{KDMD}} = \boldsymbol{\Sigma}_r^{+}\mathbf{Q}_r^{\mathrm{H}}\boldsymbol{\Psi}_{\mathbf{x}'}\mathbf{Z}_r^{\mathrm{H}} = \boldsymbol{\Sigma}_r^{+}\mathbf{Q}_r^{\mathrm{H}}\boldsymbol{\Psi}_{\mathbf{x}'}\boldsymbol{\Psi}_{\mathbf{x}}^{\mathrm{H}}\mathbf{Q}_r\boldsymbol{\Sigma}_r^{+}, \tag{2.23}$$

where $\boldsymbol{\Psi}_{\mathbf{x}}\boldsymbol{\Psi}_{\mathbf{x}}^{\mathrm{H}} = \mathbf{Q}_r\boldsymbol{\Sigma}_r^2\mathbf{Q}_r^{\mathrm{H}}$ with $(\boldsymbol{\Psi}_{\mathbf{x}}\boldsymbol{\Psi}_{\mathbf{x}}^{\mathrm{H}})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $(\boldsymbol{\Psi}_{\mathbf{x}'}\boldsymbol{\Psi}_{\mathbf{x}}^{\mathrm{H}})_{ij} = k(\mathbf{x}_{i+1}, \mathbf{x}_j)$ for $0 \leqslant i, j \leqslant M-2$. Again, such a minimizer is independent of *the choice of* $\mathbf{A}'$.

Notice that, to compute Koopman eigenvalues and eigenfunctions, one would only need access to $\boldsymbol{\Psi}_{\mathbf{x}'}\boldsymbol{\Psi}_{\mathbf{x}}^{\mathrm{H}}$ and $\boldsymbol{\Psi}_{\mathbf{x}}\boldsymbol{\Psi}_{\mathbf{x}}^{\mathrm{H}}$, i.e., the inner product among features on all data points. Fortunately, on a compact domain $\mathcal{M}$, it is well-known from Mercer's theorem (*Mercer*, 1909b) that once a suitable non-negative kernel function $k(\cdot,\cdot) : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R}$ is defined, $k(\mathbf{x}, \mathbf{y})$ is the inner product among a potentially infinite dimensional feature vector $\boldsymbol{\Psi}$ evaluated at $\mathbf{x}, \mathbf{y} \in \mathcal{M}$. Note that the choice of such a feature vector might not be unique but the corresponding reproducing kernel Hilbert space (RKHS) is (*Aronszajn*, 1950). In the case of a Gaussian kernel, one can choose the canonical feature vector $k(\cdot, \mathbf{x})$ which are "bumps" of a certain band-width distributed on $\mathcal{M}$. From the view point of kernel PCA (*Williams et al.*, 2014), $\mathbf{Q}_r$ resulting from the finite dimensional rank truncation on the Gram matrix $\boldsymbol{\Psi}_{\mathbf{x}}\boldsymbol{\Psi}_{\mathbf{x}}^{\mathrm{H}}$ is a numerical approximation to the r most dominant variance-explained mode shapes in the RKHS evaluated on the data points (*Rasmussen*, 2003), and $\mathbf{Z}_r$ represents the r dominant variance-explaining directions in terms of the feature vector in the RKHS.

Similar to EDMD, given $\hat{\mathbf{K}}_{\mathrm{KDMD}} = \hat{\mathbf{V}}\hat{\boldsymbol{\Lambda}}\hat{\mathbf{V}}^{-1}$, $\hat{\mathbf{V}} = \begin{bmatrix} \hat{\mathbf{v}}_1 \dots \hat{\mathbf{v}}_r \end{bmatrix}$, for $i = 1, \dots, r$, the corresponding i-th Koopman eigenfunctions φ_i and Koopman modes for a vector

observable $\mathbf{g}$ are,

$$\varphi_i(\mathbf{x}) = \mathbf{\Psi}(\mathbf{x})\mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}}\mathbf{Q}_r\mathbf{\Sigma}_r^{+}\hat{\mathbf{v}}_i, \tag{2.24}$$

$$\mathbf{B} = (\mathbf{\Psi}_{\mathbf{x}}\mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}}\mathbf{Q}_r\mathbf{\Sigma}_r^{+}\hat{\mathbf{V}})^{+}\begin{bmatrix}\mathbf{g}(\mathbf{x}_0)\\ \vdots \\ \mathbf{g}(\mathbf{x}_{M-1})\end{bmatrix}. \tag{2.25}$$

2.3.2 Continuous-time Formulation

To the best of our knowledge, continuous-time KDMD has not been previously reported in the literature. This can be helpful when non-uniform multi-scale samples are collected. For the kernel trick to be applied in the continuous formulation, we write $\mathbf{\Psi}_{\mathbf{x}'}$ as,

$$\mathbf{\Psi}_{\mathbf{x}'} = \begin{bmatrix}\mathbf{F}(\mathbf{x}_0)\cdot\nabla_{\mathbf{x}}\mathbf{\Psi}(\mathbf{x}_0)\\ \vdots \\ \mathbf{F}(\mathbf{x}_{M-1})\cdot\nabla_{\mathbf{x}}\mathbf{\Psi}(\mathbf{x}_{M-1})\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{x}}_0\cdot\nabla_{\mathbf{x}}\mathbf{\Psi}(\mathbf{x}_0)\\ \vdots \\ \dot{\mathbf{x}}_{M-1}\cdot\nabla_{\mathbf{x}}\mathbf{\Psi}(\mathbf{x}_{M-1})\end{bmatrix}. \tag{2.26}$$

To compute $\mathbf{\Psi}_{\mathbf{x}'}\mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}}$, denoting the q-th component of $\mathbf{F}$ as f_q,

$$\begin{aligned}
(\mathbf{\Psi}_{\mathbf{x}'}\mathbf{\Psi}_{\mathbf{x}}^{\mathrm{H}})_{ij} &= \mathbf{F}(\mathbf{x}_i)\cdot\nabla_{\mathbf{x}}\mathbf{\Psi}(\mathbf{x}_i)\mathbf{\Psi}^{\mathrm{H}}(\mathbf{x}_j)\\
&= \sum_{l=1}^{L}\sum_{q=1}^{N}\left(f_q(\mathbf{x})\frac{\partial\psi_l(\mathbf{x})}{\partial x_q}\right)\Big|_{\mathbf{x}=\mathbf{x}_i}\overline{\psi_l(\mathbf{x})}\Big|_{\mathbf{x}=\mathbf{x}_j}\\
&= \sum_{q=1}^{N} f_q(\mathbf{x}_i)\frac{\partial}{\partial x_q}\sum_{l=1}^{L}\left(\psi_l(\mathbf{x})\overline{\psi_l(\mathbf{x}')}\right)\Big|_{\mathbf{x}=\mathbf{x}_i,\mathbf{x}'=\mathbf{x}_j}\\
&= \mathbf{F}(\mathbf{x}_i)\cdot\nabla_{\mathbf{x}}k(\mathbf{x},\mathbf{x}')|_{\mathbf{x}=\mathbf{x}_i,\mathbf{x}'=\mathbf{x}_j}\\
&= \dot{\mathbf{x}}_i\cdot\nabla_{\mathbf{x}}k(\mathbf{x},\mathbf{x}')|_{\mathbf{x}=\mathbf{x}_i,\mathbf{x}'=\mathbf{x}_j},
\end{aligned} \tag{2.27}$$

where the overline symbol is the complex-conjugate operator, and the appearance of Jacobian indicates that a differentiable kernel function is required for the extension to

continuous-time. For common kernels used in Koopman analysis, the kernel function, Jacobian, and hyperparameters are listed in table 2.1.

Table 2.1: Common choice of differentiable kernel functions.

kernel type	$k(\mathbf{x}, \mathbf{x}')$	$\nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}')$	hyper para.
linear	$\mathbf{x}\mathbf{x}'^{\mathrm{H}}$	$\mathbf{x}'^{\mathrm{H}}$	
polynomial	$(1 + \mathbf{x}\mathbf{x}'^{\mathrm{H}})^{\alpha}$	$\alpha(1 + \mathbf{x}\mathbf{x}'^{\mathrm{H}})^{\alpha-1}\mathbf{x}'^{\mathrm{H}}$	α
isotropic Gaussian	$\exp(-\lVert\mathbf{x} - \mathbf{x}'\rVert^2/\sigma^2)$	$\frac{-2(\mathbf{x}-\mathbf{x}')^{\mathrm{H}}}{\sigma^2}\exp(-\lVert\mathbf{x} - \mathbf{x}'\rVert^2/\sigma^2)$	σ

2.4 Time Delay Dynamic Mode Decomposition

2.4.1 Discrete-time Formulation

Still, we consider M uniformly sampled trajectory data $\{\mathbf{x}_j\}_{j=0}^{M-1}$. The essence of time delay DMD is simply a linear model with time-delay embedding order L *assumes* that the predicted future state $\hat{\mathbf{x}}_{j+1}$ is a sum of $L+1$ linear mappings from the present state $\mathbf{x}_j$ and previous L states $\{\mathbf{x}_{j-l}\}_{l=1}^{L}$, $j \in \mathbb{N}$,

$$\hat{\mathbf{x}}_{j+1} = \mathbf{W}_0\mathbf{x}_j + \mathbf{W}_1\mathbf{x}_{j-1} + \ldots + \mathbf{W}_L\mathbf{x}_{j-L}, \tag{2.28}$$

where $\mathbf{W}_l \in \mathbb{R}^{n\times n}$ is the associated weight matrix for the l-th time-delay snapshot, $l = 0, \ldots, L$. As a side note, many data-driven models such as ERA (*Juang and Pappa*, 1985), AR, VAR (*Box et al.*, 2015), SSA (*Vautard et al.*, 1992), HAVOK (*Brunton et al.*, 2017), Hankel-DMD (*Arbabi and Mezic*, 2017) or HODMD (*Le Clainche and Vega*, 2017), can be derived from the above setup by leveraging impulse response data, introducing stochasticity, analyzing the eigenspectrum on the principal components, or adding intermittent forcing as inputs.

Given a set of M sequential snapshots: $\hat{\mathbf{x}}_{j+1}$, the goal of linear models is to determine the weight matrices that result in the best possible approximation of the

actual future state $\mathbf{x}_{j+1}$ in a priori L_2 sense, i.e.,

$$\mathbf{W}_0, \ldots, \mathbf{W}_L = \arg \min_{\{\mathcal{W}_i\}_{i=0}^{L} \in \mathbb{R}^{n \times n}} \left\| \begin{bmatrix} \mathcal{W}_L & \ldots & \mathcal{W}_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 & \ldots & \mathbf{x}_{M-2-L} \\ \vdots & \vdots & \vdots \\ \mathbf{x}_L & \ldots & \mathbf{x}_{M-2} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_{L+1} & \ldots & \mathbf{x}_{M-1} \end{bmatrix} \right\|_F, \tag{2.29}$$

if the minimizer is unique. Otherwise,

$$\mathbf{W}_0, \ldots, \mathbf{W}_L = \arg \min_{\mathcal{W}_0, \ldots, \mathcal{W}_L \in \mathbb{R}^{n \times n}} \| \begin{bmatrix} \mathcal{W}_L & \ldots & \mathcal{W}_0 \end{bmatrix} \|_F, \tag{2.30}$$

subject to

$$\begin{bmatrix} \mathcal{W}_L & \ldots & \mathcal{W}_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 & \ldots & \mathbf{x}_{M-2-L} \\ \vdots & \vdots & \vdots \\ \mathbf{x}_L & \ldots & \mathbf{x}_{M-2} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{L+1} & \ldots & \mathbf{x}_{M-1} \end{bmatrix}.$$

The analytical solution of the above optimization in Equations (2.29) and (2.30) is simply the pseudoinverse with SVD (*Schmid*, 2010), with truncation for robustness. However, straightforward SVD computation of the L time-delay matrix for large-scale dynamical systems, e.g., fluid flows $n \sim O(10^6)$ with $L \sim O(10^2)$, is challenging. Therefore, it is prudent to perform truncation in terms of spatial degree of freedom using the SVD computed from $\{\mathbf{x}_j\}_{j=0}^{M-1}$ that reduces the dimension from n to r ($r \ll n$ and $r \leqslant \min(n, M)$) and then perform the above optimizations with L time-delays on the r-dimensional system (*Le Clainche and Vega*, 2017).

2.4.2 Mode Decomposition with Time Delays

As indicated earlier, the trajectory predicted by linear models with time-delay can be viewed as constructing observable from an associated high dimensional linear system. To see this, consider a uniformly sampled trajectory data of length M, $\{\mathbf{x}_j\}_{j=0}^{M-1}$. The L time-delay vector for an n-dimensional non-linear system $\mathbf{x} \in \mathbb{R}^n$ is

defined as,

$$\mathbf{h}_k = \begin{bmatrix} \mathbf{x}_{k-L} \\ \vdots \\ \mathbf{x}_k \end{bmatrix}, \quad L \leqslant k \leqslant M-1. \tag{2.31}$$

Suppose the trajectory data can be well approximated by a linear model with L time-delays of the form in Equation (2.28). In that case, one has the so-called high order dynamic mode decomposition (*Brunton et al.*, 2017; *Le Clainche and Vega*, 2017) that approximates the non-linear dynamics well, for $L \leqslant k \leqslant M-2$,

$$\mathbf{h}_{k+1} \approx \mathbf{A}_L \mathbf{h}_k, \tag{2.32}$$

$$\mathbf{x}_{k+1} = \mathbf{E}_L \mathbf{h}_{k+1} \approx \mathbf{E}_L \mathbf{A}_L \mathbf{h}_k = \mathbf{W}_L \mathbf{x}_{k-L} + \ldots + \mathbf{W}_0 \mathbf{x}_k \tag{2.33}$$

$$\mathbf{x}_{k+1} = \mathbf{E}_L \mathbf{h}_{k+1} \approx \mathbf{E}_L \mathbf{A}_L^{k+1-L} \mathbf{h}_L = \mathbf{Q}_L \mathbf{\Lambda}^{k+1-L} \mathbf{P}_L \tag{2.34}$$

$$\mathbf{x}_{k+1} \approx \sum\nolimits_{i=1}^{n(L+1)} \lambda_i^{k+1-L} \mathbf{q}_i \mathbf{p}_i^\top \mathbf{h}_L \tag{2.35}$$

where $\mathbf{E}_L \triangleq \begin{bmatrix} \mathbf{0} & \ldots & \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{n \times n(L+1)}$, and $\mathbf{A}_L \in \mathbb{R}^{n(L+1) \times n(L+1)}$ is known as the block companion matrix,

$$\mathbf{A}_L = \begin{bmatrix} & \mathbf{I} & & & \\ & & \mathbf{I} & & \\ & & & \ddots & \\ & & & & \mathbf{I} \\ \mathbf{W}_L & \mathbf{W}_{L-1} & \mathbf{W}_{L-2} & \ldots & \mathbf{W}_0 \end{bmatrix} = \mathbf{P}_L \mathbf{\Lambda}_L \mathbf{P}_L^{-1}, \tag{2.36}$$

and

$$\mathbf{P}_L^{-1} \triangleq \begin{bmatrix} \mathbf{p}_1^\top \\ \vdots \\ \mathbf{p}_{n(L+1)}^\top \end{bmatrix}, \quad \mathbf{Q}_L \triangleq \mathbf{E}_L \mathbf{P}_L = \begin{bmatrix} \mathbf{q}_1 & \ldots & \mathbf{q}_{n(L+1)} \end{bmatrix}. \tag{2.37}$$

Note that the above decomposition in Equation (2.35) reduces to the standard

DMD when $L = 0$, i.e.,

$$\mathbf{x}_{k+1} = \sum_{i=1}^{n} \lambda_i^{k+1-L} \mathbf{q}_i \mathbf{p}_i^\top \mathbf{x}_0, \quad \forall L \leqslant k \leqslant M-2, \tag{2.38}$$

where $\mathbf{q}_i$ and $\{\lambda_i^{k+1-L} \mathbf{p}_i^\top \mathbf{x}_0\}_{k=0}^{M-2}$ are sometimes referred to as the i-th *spatial* and *temporal modes*, respectively. With more time-delays L, the maximal number of linear waves in the model increases with $n(L+1)$. As a side note, the above modal decomposition can be interpreted as an *approximation* to the Koopman mode decomposition on the trajectory with L time-delays as observables.

2.5 Deep Learning DMD

2.5.1 Neural network

The basic building block for DL-DMD is called feedforward neural network (FNN) (see appendix B.1 for more information). A typical FNN with L layers is the mapping $g(\cdot; \mathbf{W}_g) : \mathbb{R}^n \mapsto \mathbb{R}^m$ such that

$$\eta_l = \sigma_l(\eta_{l-1} W_l + b_l), \tag{2.39}$$

for $l = 1, \ldots, L-1$, where η_0 stands for the input of the neural network x, $\eta_l \in \mathbb{R}^{n_l}$ is the number of hidden units in the layer l. σ_l is activation function of layer l. Note that the last layer is linear, i.e., $\sigma_L(x) = x$:

$$g(\eta_0; \mathbf{W}_g) = \eta_{L-1} W_L + b_L, \tag{2.40}$$

where $\mathbf{W}_g = \{W_1, b_1, \ldots, W_L, b_L\}$, $W_l \in \mathbb{R}^{n_{l-1} \times n_l}$, $b_l \in \mathbb{R}^{n_l}$, for $l = 1, \ldots, L$. Note that $n_0 = n \in \mathbb{N}$, $n_L = m \in \mathbb{N}$. Such a mapping, given an arbitrary number of hidden units, even with a single hidden layer, is a universal approximator in the L^p sense $1 \leqslant p <$

∞, as long as the activation function is not polynomial almost everywhere (*Leshno et al.*, 1993). Throughout this thesis, we use the Swish (*Ramachandran et al.*, 2017) activation function, which is continuously differentiable and found to achieve strong empirical performance over many variants of ReLU and ELU on typical deep learning tasks.

2.5.2 Searching for Koopman operator as a constrained variational problem

Here we define the ideal problem of learning the Koopman decomposition as a constrained variational problem and incorporate assumptions to make it tractable step by step. Then, we introduce data as an empirical measure into the optimization in the function space. Specifically, we propose two slightly different formulations based on varying requirements.

1. Differential form: for low-dimensional systems when the governing equations, i.e., eq. (5.5) are known, they can be leveraged without the trajectory data.

2. Recurrent form: for high-dimensional systems where one can only access discrete trajectory data, which implies the access to the action of $\mathcal{K}_t$ over discrete t.

Recall that we are searching the Koopman operator defined on $\mathcal{F} = L^2(\mathcal{M}, \mu)$, which is the space of all measurable functions $\phi : \mathcal{M} \mapsto \mathbb{R}$ such that,

$$\|\phi\|_{\mathcal{F}} \triangleq \sqrt{\int_{\mathcal{M}} |\phi|^2 d\mu} < \infty. \tag{2.41}$$

As a natural extension, for any finite n, given the vector-valued function $\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \dots & \phi_n \end{bmatrix} \in \mathcal{F}^n$, we define the corresponding norm as,

$$\|\mathbf{\Phi}\|_{\mathcal{F}^n} \triangleq \sqrt{\int_{\mathcal{M}} \sum_{j=1}^{n} |\phi_j|^2 d\mu}. \tag{2.42}$$

2.5.2.1 Leveraging known physics: differential form

For any D-dimensional observation vector $\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \dots & \phi_D \end{bmatrix} \in \mathcal{F}^D$, we can define the following Koopman error functional $\mathcal{J}[\cdot] : \mathcal{F}^D \mapsto \mathbb{R}_{\geqslant 0}$, $\mathbb{R}_{\geqslant 0} = [0, +\infty)$ as the square of maximal distance for all components in Φ between the action of $\mathcal{K}$ and its L^2 projection onto $\mathcal{F}_D$ (*Korda and Mezić*, 2018a),

$$\mathcal{J}[\mathbf{\Phi}] = \max_{\psi \in \{\phi_1, \dots, \phi_D\}} \min_{f \in \mathcal{F}_D} \|f - \mathcal{K}\psi\|_{\mathcal{F}}^2, \tag{2.43}$$

which describes the extent to which $\mathcal{F}_D$ is invariant to the Koopman operator $\mathcal{K}_t$ with $t \to 0$ with respect to each basis. Ideally, if we can find $\mathbf{\Phi}$ such that the corresponding $\mathcal{F}$ is invariant to $\mathcal{K}$, i.e., $\mathcal{J}[\mathbf{\Phi}] = 0$, then $\mathcal{F}_D$ is invariant to $\mathcal{K}_t$, $\forall t > 0$, i.e., a perfect *linear* embedding in the $L^2(\mu)$ sense. Once such an embedding is determined, the realization of $\mathcal{K}$ is simply the matrix $\mathbf{K}$. In this work, we are interested in those $\mathbf{\Phi}$ such that one can recover $\mathbf{x}$ and $\mathcal{J}[\mathbf{\Phi}] \geqslant 0$. Note that if $\mathbf{\Phi}$ is restricted to linear mapping, then the resulting $\mathbf{\Phi}$ spans a D-dimensional Koopman invariant subspace that also contains the system state $\mathbf{x}$, which can be very helpful for optimal control. Otherwise, if $\mathbf{\Phi}$ becomes non-linear, then we refer it as *searching a linear embedding* of the non-linear dynamical system, which can be efficient in modeling and prediction of the dynamical system (*Otto and Rowley*, 2019b). However, since one can't reconstruct the system state linearly, the concept of Koopman modes in eq. (1.13) is not applicable anymore.

Although the above problem setup only contains minimal necessary assumptions, it is both mathematically and computationally challenging. For practical purposes, we consider the following assumptions to make the problem tractable:

1. Instead of solving the equation $\mathcal{J}[\mathbf{\Phi}] = 0$, we search for $\mathbf{\Phi}$ by finding the

minimum of the following constrained variational problem,

$$\boldsymbol{\Phi}^* = \underset{\substack{\boldsymbol{\Phi}\in\mathcal{F}^D,\exists\boldsymbol{\Psi}:\mathbb{R}^D\mapsto\mathcal{M}\\ \boldsymbol{\Psi}\circ\boldsymbol{\Phi}=\mathcal{I}}}{\arg\min}\ \mathcal{J}[\boldsymbol{\Phi}]. \tag{2.44}$$

2. Instead of directly solving the variational problem in the infinite-dimensional $\mathcal{F}^D$, we optimize $\boldsymbol{\Phi}$ in the finite-dimensional parameter space of feedforward neural networks with fixed architecture. Note that the number of layers and the number of hidden units in each layer are determined heuristically. Thus, we are searching in a subset of $\mathcal{F}^D$ described by $\mathbf{W}_{\boldsymbol{\Phi}}$, which might induce a gap due to the choice of the neural network architecture,

$$0 \leqslant \min_{\substack{\boldsymbol{\Phi}\in\mathcal{F}^D,\exists\boldsymbol{\Psi}:\mathbb{R}^D\mapsto\mathcal{M}\\ \boldsymbol{\Psi}\circ\boldsymbol{\Phi}=\mathcal{I}}} \mathcal{J}[\boldsymbol{\Phi}] \leqslant \min_{\substack{\mathbf{W}_{\boldsymbol{\Phi}},\exists\boldsymbol{\Psi}\in C(\mathbb{R}^D,\mathbb{R}^N)\\ \boldsymbol{\Psi}\circ\boldsymbol{\Phi}(\cdot;\mathbf{W}_{\boldsymbol{\Phi}})=\mathcal{I}}} \mathcal{J}[\boldsymbol{\Phi}(\cdot;\mathbf{W}_{\boldsymbol{\Phi}})]. \tag{2.45}$$

Clearly, the gap is bounded by the right-hand side of the second inequality above. In addition, it should be noted that the requirement of linear independence in $\{\phi_1,\ldots,\phi_D\}$ is relaxed, but $\dim\mathcal{F}_D$ is bounded by D.

3. As a standard procedure in deep learning (*Goodfellow et al.*, 2016), we use the penalty method to *approximate* the constrained optimization problem with an unconstrained optimization with unity penalty coefficient. Since this still entails nonconvex optimization, we define a global minimum as follows:

$$\mathbf{W}^*_{\boldsymbol{\Phi}} = \underset{\substack{\mathbf{W}_{\boldsymbol{\Phi}},\exists\boldsymbol{\Psi}\in C(\mathbb{R}^D,\mathbb{R}^N)\\ \boldsymbol{\Psi}\circ\boldsymbol{\Phi}(\cdot;\mathbf{W}_{\boldsymbol{\Phi}})=\mathcal{I}}}{\arg\min}\ \mathcal{J}[\boldsymbol{\Phi}(\cdot;\mathbf{W}_{\boldsymbol{\Phi}})], \tag{2.46}$$

$$\widehat{\mathbf{W}}_{\boldsymbol{\Phi}}, \widehat{\mathbf{W}}_{\boldsymbol{\Psi}} = \underset{\mathbf{W}_{\boldsymbol{\Phi}},\mathbf{W}_{\boldsymbol{\Psi}}}{\arg\min}\ \mathcal{J}[\boldsymbol{\Phi}(\cdot;\mathbf{W}_{\boldsymbol{\Phi}})] + \mathcal{R}[\boldsymbol{\Phi}(\cdot;\mathbf{W}_{\boldsymbol{\Phi}}),\boldsymbol{\Psi}(\cdot;\mathbf{W}_{\boldsymbol{\Psi}})], \tag{2.47}$$

where the reconstruction error functional $\mathcal{R}[\cdot,\cdot] : \mathcal{F}^D \times C(\mathbb{R}^D,\mathbb{R}^N) \mapsto \mathbb{R}_{\geqslant 0}$, is

defined as

$$\mathcal{R}[\mathbf{\Phi}, \mathbf{\Psi}] = \|\mathbf{\Psi} \circ \mathbf{\Phi} - \mathcal{I}\|^2_{\mathcal{F}^N}\,, \tag{2.48}$$

for $\mathbf{\Phi} \in \mathcal{F}^D$, $\mathbf{\Psi} \in C(\mathbb{R}^D, \mathbb{R}^N)$. Then we assume $\mathbf{\Phi}(\cdot; \widehat{\mathbf{W}}_{\mathbf{\Phi}})$ approximates one of the global minima, i.e., $\mathbf{\Phi}(\cdot; \mathbf{W}^*_{\mathbf{\Phi}})$. Note that the convergence to a global minimum of the constrained optimization can be guaranteed if one is given the global minima of the sequential unconstrained optimization and increasing the penalty coefficient to infinity (*Luenberger*, 1973).

4. We then optimize the sum of square error $\widehat{\mathcal{J}}$ overall components of $\mathbf{\Phi}$, which serves an upper bound of $\mathcal{J}$,

$$\mathcal{J}[\mathbf{\Phi}] \leqslant \widehat{\mathcal{J}}[\mathbf{\Phi}] = \sum_{i=1}^{D} \min_{f \in \mathcal{F}_D} \|f - \mathcal{K}\phi_i\|^2_{\mathcal{F}} = \min_{\mathbf{K} \in \mathbb{R}^{D\times D}} \widetilde{\mathcal{J}}[\mathbf{\Phi}, \mathbf{K}], \tag{2.49}$$

$$\widetilde{\mathcal{J}}[\mathbf{\Phi}, \mathbf{K}] = \|\mathbf{\Phi}\mathbf{K} - \mathcal{K}\mathbf{\Phi}\|^2_{\mathcal{F}^D}\,. \tag{2.50}$$

The above formulation also implies equal importance among all components of $\mathbf{\Phi}$.

5. Despite the non-convex nature of the problem, we employ gradient-descent optimization to search for a local minimum.

In summary, based on the above assumptions, we will solve the following optimization problem by gradient-descent:

$$\widehat{\mathbf{W}}_{\mathbf{\Phi}}, \widehat{\mathbf{W}}_{\mathbf{\Psi}}, \widehat{\mathbf{K}} = \underset{\mathbf{W}_{\mathbf{\Phi}}, \mathbf{W}_{\mathbf{\Psi}}, \mathbf{K}}{\arg\min}\, \widetilde{\mathcal{J}}[\mathbf{\Phi}(\cdot; \mathbf{W}_{\mathbf{\Phi}}), \mathbf{K}] + \mathcal{R}[\mathbf{\Phi}(\cdot; \mathbf{W}_{\mathbf{\Phi}}), \mathbf{\Psi}(\cdot; \mathbf{W}_{\mathbf{\Psi}})]. \tag{2.51}$$

2.5.2.2 Unknown physics with only the trajectory data: recurrent form

From the viewpoint of approximation, it can be argued that the most natural way to determine the continuous Koopman operator is in the aforementioned differential form. However, higher accuracy can be achieved by taking advantage of trajectory

information and minimizing the error over multiple time steps in the trajectory. This is the key idea behind optimized DMD (*Askham and Kutz*, 2018). Recently, *Lusch et al.* (2018a), *Otto and Rowley* (2019a) extended this idea to determine the discrete-time Koopman operator using deep learning.

Recall that we assume that the space of observation functions $\mathcal{F} = L^2(\mathcal{M}, \mu)$ is invariant to $\mathcal{K}_t$, $\forall t \in \mathbb{R}_{\geqslant 0}$. Thus, we consider the t-time evolution of any function $\phi \in \mathcal{F}$, i.e., $\mathcal{K}_t\phi(\mathbf{x})$, as a function on $\mathcal{U} = \mathcal{M} \times \mathcal{T}$, given an initial condition $\mathbf{x} \in \mathcal{M}$ and time evolution $t \in \mathcal{T}$ where $\mathcal{T}$ is a one dimensional smooth manifold, sometimes also referred to as the time horizon. Recall that we assume $\mathcal{K}_t\phi(\mathbf{x}) \in \mathcal{G} \triangleq L^2(\mathcal{U}, \nu)$. Based on the fact that $\mathcal{F}$ is invariant to $\mathcal{K}_t$, such an assumption can be shown to be valid for compact $\mathcal{T}$ for a proper measure ν.

Similar to the differential form, for any observation vector $\mathbf{\Phi} \in \mathcal{F}^D$, we define the following Koopman error functional $\mathcal{J}_r[\cdot] : \mathcal{F}^D \mapsto \mathbb{R}_{\geqslant 0}$ as the square of maximal L^2 distance for all components in $\mathbf{\Phi}$ between the predicted and ground truth trajectory,

$$\mathcal{J}_r[\mathbf{\Phi}] = \max_{\psi \in \{\phi_1, \ldots, \phi_D\}} \min_{\mathbf{K} \in \mathbb{R}^{D \times D}} \left\| \mathbf{\Phi} e^{t\mathbf{K}} c_\psi - \mathcal{K}_t \psi \right\|_{\mathcal{G}}^2, \tag{2.52}$$

where $\psi = \mathbf{\Phi} c_\psi$, $c_\psi \in \mathbb{R}^{D \times 1}$. Following similar assumptions in section 2.5.2.1, we need to define a reconstruction error functional to describe the discrepancy between the reconstructed and original states. Indeed, one can directly define the prediction error functional in the recurrent form, $\widetilde{\mathcal{P}}[\cdot, \cdot, \cdot] : \mathcal{F}^D \times C(\mathbb{R}^D, \mathbb{R}^N) \times \mathbb{R}^{D \times D} \mapsto \mathbb{R}_{\geqslant 0}$ as

$$\widetilde{\mathcal{P}}[\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{K}] = \left\| \mathbf{\Psi} \circ \mathbf{\Phi} e^{t\mathbf{K}} - \mathcal{K}_t \mathcal{I} \right\|_{\mathcal{G}^N}^2. \tag{2.53}$$

Similarly, we define

$$\widetilde{\mathcal{J}}_r[\mathbf{\Phi}(\cdot; \mathbf{W}_{\mathbf{\Phi}}), \mathbf{K}] = \left\| \mathbf{\Phi} e^{t\mathbf{K}} - \mathcal{K}_t \mathbf{\Phi} \right\|_{\mathcal{G}^D}^2, \tag{2.54}$$

and solve the following optimization problem via a gradient-based algorithm:

$$\widehat{\mathbf{W}}_{\mathbf{\Phi}}, \widehat{\mathbf{W}}_{\mathbf{\Psi}}, \widehat{\mathbf{K}} = \underset{\mathbf{W}_{\mathbf{\Phi}}, \mathbf{W}_{\mathbf{\Psi}}, \mathbf{K}}{\arg\min}\, \widetilde{\mathcal{J}}_r[\mathbf{\Phi}(\cdot; \mathbf{W}_{\mathbf{\Phi}}), \mathbf{K}] + \widehat{\mathcal{P}}[\mathbf{\Phi}(\cdot; \mathbf{W}_{\mathbf{\Phi}}), \mathbf{\Psi}(\cdot; \mathbf{W}_{\mathbf{\Psi}}), \mathbf{K}]. \tag{2.55}$$

As a side note, one might also define the following reconstruction functional similar to previous differential form that is independent of $\mathbf{K}$,

$$\mathcal{R}[\mathbf{\Phi}, \mathbf{\Psi}] = \|\mathcal{K}_t(\mathbf{\Psi} \circ \mathbf{\Phi} - \mathcal{I})\|^2_{\mathcal{G}^N}\,, \tag{2.56}$$

which can bound the prediction error functional together with the Koopman error functional by triangular inequality,

$$\begin{aligned} \left\|\mathbf{\Psi} \circ \mathbf{\Phi} e^{t\mathbf{K}} - \mathcal{K}_t \mathcal{I}\right\|_{\mathcal{G}^N} &\leqslant \left\|\mathbf{\Psi} \circ \mathbf{\Phi} e^{t\mathbf{K}} - \mathcal{K}_t \mathbf{\Psi} \circ \mathbf{\Phi}\right\|_{\mathcal{G}^N} + \|\mathcal{K}_t(\mathbf{\Psi} \circ \mathbf{\Phi} - \mathcal{I})\|_{\mathcal{G}^N}\,, \quad (2.57) \\ &\leqslant L_{\mathbf{\Psi}} \left\|\mathbf{\Phi} e^{t\mathbf{K}} - \mathcal{K}_t \mathbf{\Phi}\right\|_{\mathcal{G}^D} + \|\mathcal{K}_t(\mathbf{\Psi} \circ \mathbf{\Phi} - \mathcal{I})\|_{\mathcal{G}^N}\,, \quad (2.58) \end{aligned}$$

where $L_{\mathbf{\Psi}}$ is the Lipschitz constant for $\mathbf{\Psi}$. However, in this work, we do not have control over $L_{\mathbf{\Psi}}$, and thus we prefer to directly minimize the prediction error function as in previous studies (*Otto and Rowley*, 2019a; *Lusch et al.*, 2018a).

2.5.3 Measures

2.5.3.1 Measure for data generated by sampling in phase space

We consider the situation where M data points on $\mathcal{M}$, i.e., $\{\mathbf{x}_m\}_{m=1}^M$ are drawn independently from some measure μ on $\mathcal{M}$, e.g., uniform distribution. This induces the following empirical measure $\hat{\mu}_M$,

$$\hat{\mu}_M = \frac{1}{M} \sum_{m=1}^{M} \delta_{\mathbf{x}_m}, \tag{2.59}$$

where δ_x is the Dirac measure for x. Note that $\hat{\mu}_M$ uniformly converges to μ (*Vapnik and Chervonenkis*, 2015) as $M \to \infty$. Thus, one can rewrite the differential form in eq. (2.51) as an empirical risk minimization (*Vapnik*, 1992),

$$
\begin{aligned}
\widehat{\mathbf{W}}_{\mathbf{\Phi}}, \widehat{\mathbf{W}}_{\mathbf{\Psi}}, \widehat{\mathbf{K}} &= \lim_{M\to\infty} \widehat{\mathbf{W}}_{\mathbf{\Phi},M}, \widehat{\mathbf{W}}_{\mathbf{\Psi},M}, \widehat{\mathbf{K}}_M \qquad (2.60)\\
&= \lim_{M\to\infty} \underset{\mathbf{W}_{\mathbf{\Phi}},\mathbf{W}_{\mathbf{\Psi}},\mathbf{K}}{\arg\min}\ \widetilde{\mathcal{J}}_M[\mathbf{\Phi}(\cdot;\mathbf{W}_{\mathbf{\Phi}}),\mathbf{K}] + \mathcal{R}_M[\mathbf{\Phi}(\cdot;\mathbf{W}_{\mathbf{\Phi}}),\mathbf{\Psi}(\cdot;\mathbf{W}_{\mathbf{\Psi}})]\\
&= \lim_{M\to\infty} \underset{\mathbf{W}_{\mathbf{\Phi}},\mathbf{W}_{\mathbf{\Psi}},\mathbf{K}}{\arg\min}\ \|\mathbf{\Phi}(\cdot;\mathbf{W}_{\mathbf{\Phi}})\mathbf{K} - \mathcal{K}\mathbf{\Phi}\|^2_{\widehat{\mathcal{F}}^D_M} + \|\mathbf{\Psi}(\cdot;\mathbf{W}_{\mathbf{\Psi}}) \circ \mathbf{\Phi}(\cdot;\mathbf{W}_{\mathbf{\Phi}}) - \mathcal{I}\|^2_{\widehat{\mathcal{F}}^N_M}\\
&= \lim_{M\to\infty} \underset{\mathbf{W}_{\mathbf{\Phi}},\mathbf{W}_{\mathbf{\Psi}},\mathbf{K}}{\arg\min}\ \frac{1}{M}\sum_{m=1}^{M}\Bigg(\|\mathbf{\Phi}(\mathbf{x}_m;\mathbf{W}_{\mathbf{\Phi}})\mathbf{K} - \mathbf{F}\cdot\nabla_{\mathbf{x}}\mathbf{\Phi}(\mathbf{x}_m)\|^2\\
&\qquad\qquad + \|\mathbf{\Psi}(\mathbf{\Phi}(\mathbf{x}_m;\mathbf{W}_{\mathbf{\Phi}});\mathbf{W}_{\mathbf{\Psi}}) - \mathbf{x}_m\|^2\Bigg),
\end{aligned}
$$

where $\widehat{\mathcal{F}}_M = L^2(\mathcal{M}, \hat{\mu}_M)$.

2.5.3.2 Measure for trajectory data generated by solving the initial value problem

In the general case, information content in trajectory data resulting from the solution of the initial value problem is strongly dependent on the initial condition. For instance, when the initial condition is in a region of phase space with sharp changes, it would be sensible to use a high sampling rate. On the other hand, if the initial condition is near a fixed point attractor, one might prefer to stop collecting the data after the system arrives at the fixed point. Such a sampling pattern for the specific initial state can be summarized as a Markov kernel (*Klenke*, 2013) $\kappa : \mathcal{M} \times \Sigma_{\mathcal{T}} \mapsto [0,1]$. For every fixed initial state $\mathbf{x} \in \mathcal{M}$, the map $\kappa(\mathcal{E}_{\mathcal{T}}, \mathbf{x})$ is a measure on $(\mathcal{T}, \Sigma_{\mathcal{T}})$, $\mathcal{E}_{\mathcal{T}} \in \Sigma_{\mathcal{T}}$, where $\Sigma_{\mathcal{T}}$ is the σ-algebra on $\mathcal{T}$. Then, there exists a unique measure ν on $\mathcal{U}$ (*Klenke*, 2013) such that,

$$
\nu(\mathcal{E}_{\mathcal{M}} \times \mathcal{E}_{\mathcal{T}}) = \int_{\mathcal{E}_{\mathcal{M}}} \kappa(\mathcal{E}_{\mathcal{T}}, \mathbf{x}) d\mu(\mathbf{x}), \qquad (2.61)
$$

for $\mathcal{E}_{\mathcal{T}} \in \Sigma_{\mathcal{T}}, \mathcal{E}_{\mathcal{M}} \in \Sigma_{\mathcal{M}}$.

Assume we are given M trajectories, $\{\{\mathbf{x}_{m,j}\}_{j=1}^{T_m}\}_{m=1}^{M}$, $\{\{t_{m,j}\}_{j=1}^{T_m}\}_{m=1}^{M}$. The initial condition for each trajectory is drawn independently from μ. For the m-th trajectory with initial condition $\mathbf{x}_m$, there are T_m samples drawn independently from measure $\kappa(\cdot, \mathbf{x}_{m,1})$ where the time elapse of j-th sample away from initial condition $\mathbf{x}_{m,1}$ is $t_{m,j}$ ($t_{m,1} = 0$), where $m = 1, \ldots, M$. Then, we can define the corresponding empirical measure from eq. (2.61),

$$\hat{\nu}_{M,\widehat{T}} = \frac{1}{M} \sum_{m=1}^{M} \delta_{\mathbf{x}_m} \left(\frac{1}{T_m} \sum_{j=1}^{T_m} \delta_{t_{m,j}} \right), \tag{2.62}$$

where $\widehat{T} = \max_{m=1,\ldots,M} T_m$, and $\hat{\nu}_{M,\widehat{T}}$ uniformly converges to ν as $M, \widehat{T} \to \infty$.

Similarly, we can rewrite eq. (2.55) as the following,

$$\begin{aligned}
\widehat{\mathbf{W}}_{\mathbf{\Phi}}, \widehat{\mathbf{W}}_{\mathbf{\Psi}}, \widehat{\mathbf{K}} &= \lim_{M,T\to\infty} \widehat{\mathbf{W}}_{\mathbf{\Phi},M,\widehat{T}}, \widehat{\mathbf{W}}_{\mathbf{\Psi},M,\widehat{T}}, \widehat{\mathbf{K}}_{M,\widehat{T}} \qquad (2.63)\\
&= \lim_{M,\widehat{T}\to\infty} \operatorname*{arg\,min}_{\mathbf{W}_{\mathbf{\Phi}},\mathbf{W}_{\mathbf{\Psi}},\mathbf{K}} \widetilde{\mathcal{J}}_{r,M,\widehat{T}}[\mathbf{\Phi}(\cdot;\mathbf{W}_{\mathbf{\Phi}}), \mathbf{K}] + \widehat{\mathcal{P}}_{M,\widehat{T}}[\mathbf{\Phi}(\cdot;\mathbf{W}_{\mathbf{\Phi}}), \mathbf{\Psi}(\cdot;\mathbf{W}_{\mathbf{\Psi}}), \mathbf{K}]\\
&= \lim_{M,\widehat{T}\to\infty} \operatorname*{arg\,min}_{\mathbf{W}_{\mathbf{\Phi}},\mathbf{W}_{\mathbf{\Psi}},\mathbf{K}} \left\| \mathbf{\Phi} e^{t\mathbf{K}} - \mathcal{K}_t \mathbf{\Phi} \right\|^2_{\hat{\mathcal{G}}_{M,\widehat{T}}} + \left\| \mathbf{\Psi} \circ \mathbf{\Phi} e^{t\mathbf{K}} - \mathcal{K}_t \mathcal{I} \right\|^2_{\hat{\mathcal{G}}^N_{M,\widehat{T}}}\\
&= \lim_{M,\widehat{T}\to\infty} \operatorname*{arg\,min}_{\mathbf{W}_{\mathbf{\Phi}},\mathbf{W}_{\mathbf{\Psi}},\mathbf{K}} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{T_m} \Bigg(\sum_{j=2}^{T_m} \| \mathbf{\Phi}(\mathbf{x}_{m,1};\mathbf{W}_{\mathbf{\Phi}}) e^{t_{m,j}\mathbf{K}} - \mathbf{\Phi}(\mathbf{x}_{m,j}) \|^2\\
&\qquad + \sum_{j=1}^{T_m} \left\| \mathbf{\Psi}(\mathbf{\Phi}(\mathbf{x}_{m,1};\mathbf{W}_{\mathbf{\Phi}}) e^{t_{m,j}\mathbf{K}}; \mathbf{W}_{\mathbf{\Psi}}) - \mathbf{x}_{m,j} \right\|^2 \Bigg),
\end{aligned}$$

where $\widehat{G} = L^2(\mathcal{U}, \widehat{\nu}_{M,\widehat{T}})$.Note that eq. (2.63) generalizes the LRAN model with a simpler loss function (*Otto and Rowley*, 2019a) and the discrete spectrum model in the paper of Lusch et al. (*Lusch et al.*, 2018a).

Note that the above data setup also generalizes to cases where data along a single long trajectory is "Hankelized" (*Arbabi and Mezic*, 2017), i.e., dividing a long trajectory into several smaller-sized consecutive trajectories. This would lead to a

truncated time horizon in the model, which leads to better computational efficiency compared to the original data, at the cost of some loss in prediction accuracy.

2.6 Summary

In this chapter, we provided a detailed mathematical description of three different classes of methods in approximating Koopman operators: EDMD/KDMD, Time Delay DMD, Deep Learning DMD. EDMD/KDMD relies on approximating the orthogonal projection of the Koopman operator with explicit/implicit dictionary functions. Time Delay DMD takes the historical information as augmented observables for approximating the Koopman operator. Deep Learning DMD parameterizes the dictionary functions by the neural network. Therefore, it becomes an approximation of a constrained variational problem. Moreover, one can further make the network recurrent to increase accuracy for long-time prediction.

In the following chapters, we will discuss the remaining technical issues of these methods and present our contributions towards discovering *accurate*, *informative* Koopman operator with *uncertainty quantification* in resolving several issues of these existing techniques and providing *theoretical*, *numerical insights* into the mechanism of these methods.

CHAPTER III

Structure of Delay Embedding in Time Delay Dynamic Mode Decomposition

> We are not makers of history. We are made by history.
>
> *Martin Luther King Jr. (1963)*

3.1 Background and Motivation

Time delay embedding naturally arises in the representation of the evolution of partially observed states in dynamical systems. As an illustrative example, consider a n-dimensional linear autonomous discrete dynamical system with Q partially observed (or resolved) states, $Q < n$:

$$\mathbf{x}_{j+1} \triangleq \begin{bmatrix} \hat{\mathbf{x}}_{j+1} \\ \tilde{\mathbf{x}}_{j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{j} \\ \tilde{\mathbf{x}}_{j} \end{bmatrix}, \tag{3.1}$$

where $\hat{\mathbf{x}}_j \in \mathbb{R}^Q$, $\tilde{\mathbf{x}}_j \in \mathbb{R}^{n-Q}$, $j \in \mathbb{N}$, $\mathbf{A}_{11} \in \mathbb{R}^{Q\times Q}$, $\mathbf{A}_{12} \in \mathbb{R}^{Q\times(n-Q)}$, $\mathbf{A}_{21} \in \mathbb{R}^{(n-Q)\times Q}$, $\mathbf{A}_{22} \in \mathbb{R}^{(n-Q)\times(n-Q)}$. The dynamical evolution of the observed states $\hat{\mathbf{x}}$ is given by:

$$\hat{\mathbf{x}}_{j+1} = \mathbf{A}_{11}\hat{\mathbf{x}}_j + \sum_{k=0}^{j-1} \mathbf{A}_{12}\mathbf{A}_{22}^k\mathbf{A}_{21}\hat{\mathbf{x}}_{j-1-k} + \mathbf{A}_{12}\mathbf{A}_{22}^j\tilde{\mathbf{x}}_0. \tag{3.2}$$

Typically, the last term decays rapidly, and thus the above equation can be considered to be *closed* in the observed variables $\hat{\boldsymbol{x}}$. The second term on the right-hand side of Equation (3.2) describes how the time-history of the observed modes affects the dynamics. Thus, Equation (3.2) implies that it is possible to extract the dynamics of the observables $\hat{\mathbf{x}}$ using time delayed observables, i.e., $\hat{\mathbf{x}}_{j+1} = \mathbf{C}_0\hat{\mathbf{x}}_j + \sum_{k=1}^{L} \mathbf{C}_k\hat{\mathbf{x}}_{j-k}$, where $\mathbf{C}_k \in \mathbb{R}^{Q\times Q}$, and L is the number of time delays. It should, however, be noted that *explicit* delays might not be necessary if one has access to high order time derivatives (*Takens*, 1981) or abundant distinct observations (*Deyle and Sugihara*, 2011).

Theorem III.1 (1981 Takens' embedding theorem)**.** *Consider a compact manifold M of dimension m, for pairs (φ, y), $\varphi : M \mapsto M$, a smooth diffeomorphism and $y : M \mapsto \mathbb{R}$ is a smooth function, it is a generic property that the map $\Phi_{(\varphi,y)} : M \mapsto \mathbb{R}^{2m+1}$, defined by*

$$\Phi_{(\varphi,y)} = [y(x) \quad y(\varphi(x)) \quad y(\varphi^2(x)) \quad \dots \quad y(\varphi^{2m}(x))]$$

is an embedding, where "smooth" means C^2, i.e., continuously differentiable.

Leveraging delay coordinates to construct predictive models of dynamical systems has been a topic of great interest. Such models have been studied extensively in the time series analysis community via the well-known family of autoregressive and moving average (ARMA) models (*Box et al.*, 2015). In a dynamical systems context, time delay embedding is leveraged in higher order or Hankel Dynamic Mode Decomposition (*Le Clainche and Vega*, 2017; *Arbabi and Mezic*, 2017; *Brunton et al.*, 2017).

A relevant and outstanding question in each of the contexts above is the following: *Given time series data from a non-linear dynamical system, how much memory is required to accurately recover the underlying dynamics, given a model structure?*

The memory can be characterized by the two hyperparameters, namely *the number of time delays* and the *corresponding data sampling intervals if uniformly sampled*. Takens embedding theorem (*Takens*, 1981) proved the generic existence of a time delayed system with $L = \lceil 2n_{box} \rceil$ delays, where (n_{box} is box-counting dimension of the attractor, given the model has enough *non-linearity* to approximate the diffeomorphism. However, the question of how to determine the number of time delays and the sampling rate is not well-addressed. Given n_{box} as the box-counting dimension of the attractor, the number of required time delays $L_{takens} = \lceil 2n_{box} \rceil$ is rather *conservative* (*Gilmore and Lefranc*, 2011). For example, it is both well known in practice and shown analytically (*Pan and Duraisamy*, 2018a) that a typical chaotic Lorenz attractor with box-counting dimension ≈ 2.06 (*McGuinness*, 1983) can be well embedded with $L = 2$, i.e., an equivalent 3D time delay system. Meanwhile, $L = 4$ is required from Takens embedding theorem.

However, other than acknowledging a diffeomorphism, Takens embedding theorem does not posit any constraints on the mapping from time delay coordinates to the original system state. Clearly, the required number of time delays depends on the richness (non-linearity) of the embedding. In general, for non-linear models, the determination of the time delays becomes a problem of phase-space reconstruction (*Frank et al.*, 2001; *Abarbanel et al.*, 1993). Popular methods include the false nearest neighbor method (*Kennel et al.*, 1992), singular value analysis (*Broomhead and Jones*, 1989), averaged mutual information (*Sugihara et al.*, 1990), saturation of system invariants (*Abarbanel et al.*, 1993), box-counting methods (*Sauer and Yorke*, 1993), correlation integrals (*Kim et al.*, 1999), standard model selection techniques (*Cao*, 1997), and even reinforcement learning (*Liu et al.*, 2007). On the other hand, for linear models, criteria based on statistical significance such as the model utility F-test (*Lomax and Hahs-Vaughn*, 2013) or information-theoretic techniques such as AIC/BIC (*Box et al.*, 2015) are used. The use of the partial autocorrelation in linear

autoregressive (AR) models to determine the number of delays can be categorized as a model selection approach. It should be mentioned that by treating the models as a black-box, a general approach such as cross validation can be leveraged.

From the viewpoint of discovering the dynamics of a partially observed system, the goal is to determine the non-linear convolution operator (*Chorin and Hald*, 2014; *Gouasmi et al.*, 2017) or the so-called closure dynamics (*Pan and Duraisamy*, 2018a). It has to be recognized that the number of time delays will also be dependent on the specific structure of the model. The interchangeability between the number of distinct observables and the number of time delays is also reflected in Takens' original work on the embedding theorem (*Takens*, 1981). Such interchangeability with the latent space dimension is also explored in closure dynamics (*Pan and Duraisamy*, 2018a; *Gouasmi et al.*, 2017; *Parish et al.*, 2018) and recurrent neural networks (*Goodfellow et al.*, 2016). Since the required number of delays is strongly dependent on the model structure, it is prudent to first narrow down to a specific type of model, and then determine the delays needed.

The connection between time delay embedding and the Koopman operator is elucidated by *Brunton et al.* (2017). Further theoretical investigations were conducted by *Arbabi and Mezic* (2017). For an ergodic dynamical system, assuming that the observable belongs to a finite-dimensional Koopman invariant subspace $\mathcal{H}$, they showed that Hankel-DMD, a linear model (first proposed and connected to ERA (*Juang and Pappa*, 1985)/SSA (*Vautard et al.*, 1992) by *Tu et al.* (2013)), can provide an exact representation of the Koopman eigenvalues and eigenfunctions in $\mathcal{H}$. This pioneering work, together with several numerical investigations on the application of Hankel-DMD to non-linear dynamical systems (*Champion et al.*, 2019; *Le Clainche and Vega*, 2017; *Brunton et al.*, 2017) and theoretical studies on time-delayed observables using singular value decomposition (SVD) (*Kamb et al.*, 2018) highlight the ability of linear time delayed models to represent non-linear dynamics. From a heuris-

tic viewpoint, SVD has been demonstrated (*Broomhead and Jones*, 1989; *Broomhead and King*, 1986; *Gibson et al.*, 1992) to serve as a practical guide to determine the required number of time delays and sampling rate, for linear models.

It should be noted that much of the literature (*Tu et al.*, 2013; *Schmid*, 2010; *Brunton et al.*, 2013) related to DMD and Hankel-DMD consider SVD projection either in the time delayed dimension (e.g. singular spectrum analysis) or the state dimension. SVD can provide optimal linear coordinates to maximize signal-to-noise ratio (*Gibson et al.*, 1992), and thus promote robustness and efficiency. On the other hand, projection via Fourier transformation enables the possibility of additional theoretical analysis. For instance, Fourier-based analysis of the Navier–Stokes equations include non-linear triadic wave interactions (*Pope*, 2000) and decomposition into solenoidal and dilatational components (*Pan and Johnsen*, 2017). Pertinent to the present thesis, ergodic systems characterized by periodic or quasi-periodic attractors have been shown to be well approximated by Fourier analysis (*Schilder et al.*, 2006; *Rowley et al.*, 2009; *Mezić*, 2005).

In this chapter, our goal is to leverage a Fourier basis representation to uncover the structure of time delay embeddings in *linear models* (described in section 2.4) of non-linear dynamical systems. Moreover, we will also address related issues of numerical conditioning and robustness to noises. We will conclude with several examples that demonstrate our findings.

3.2 Problem Setup and Assumptions

Recall that we are interested in an n-dimensional continuous autonomous non-linear dynamical system,

$$\frac{d}{dt}\mathbf{x} = \mathbf{F}(\mathbf{x}), \tag{3.3}$$

on a state space $\mathcal{M} \subset \mathbb{R}^n$, $n \in \mathbb{N}^+$. The initial condition is denoted as $\mathbf{x}(0) = \mathbf{x}_0$. By uniformly sampling with time interval Δt, the trajectory data of the dynamical systems can be obtained as $\{\mathbf{x}_j\}_{j=0}^{\infty}$, where $\mathbf{x}_j \triangleq \mathbf{x}(j\Delta t), j \in \mathbb{N}$. This is our setup for the data with the time delay DMD previously described in section 2.4.

To provide insight into role of time-delays, we consider the following simplification for the ease of analysis: we restrict ourselves to the dynamics on a periodic attractor, for which one can determine an arbitrarily close Fourier interpolation in time at a uniform sampling rate (*Attinger et al.*, 1966). In addition, without loss of generality, we assume that the data has zero mean, i.e., $\int_{\mathbb{R}^+} \mathbf{x}(\tau) d\tau = \mathbf{0}$.

3.3 Theory

3.3.1 Scalar Time Series

Here we start the theoretical analysis with the scalar case i.e., $n = 1$. Then we will extend to the vector case $n > 1$. Note that the corresponding data is collected by uniformly sampling a T-periodic time series $x(t) \in \mathbb{R}$. The number of samples per period is M, with uniform sampling interval $\Delta t = T/M$. Without loss of generality, we assume that sampling is initiated at $t = 0$, $x_k = x(t_k)$, $t_k = k\Delta t$, $k \in \mathcal{I}_M$, $\mathcal{I}_M = \{0, 1, \ldots, M-1\}$, and T is the smallest positive real number that represents the periodicity.

3.3.1.1 Projection on Fourier basis

With the above simplifications, we consider a surrogate signal of $x(t)$ as $S_M(t)$ defined in eq. (3.4).

$$S_M(t) = \sum_{i \in \mathcal{I}_M} a_i e^{-j\frac{2\pi i t}{T}} \quad \text{with} \quad a_i = \frac{1}{M} \sum_{k \in \mathcal{I}_M} x_k e^{j\frac{2\pi k i}{M}} \in \mathbb{C}, \tag{3.4}$$

where $j = \sqrt{-1}$ and

$$\forall k \in \mathcal{I}_M, \quad x_k = x(k\Delta t) = S_M(k\Delta t), \tag{3.5}$$

which is obtained by projecting $\mathbf{x}(t)$ on the following linear space $\mathcal{H}_F$

$$\mathcal{H}_F = \text{span}\{1, e^{-j\frac{2\pi t}{T}}, \ldots, e^{-j\frac{2\pi(M-1)t}{T}}\}, \tag{3.6}$$

which is spanned by the Fourier basis in Equation (3.6) with test functions as delta functions as $\delta(t-t_k), k \in \mathcal{I}_M$. This step is equivalent to the discrete Fourier transform (DFT).

The above procedure naturally represents the uniformly sampled trajectory in the time domain $\{x_k\}_{k=0}^{M-1}$ using coefficients in the frequency domain $\{a_i\}_{i=0}^{M-1}$. Since we consider real signals, $\{a_i\}_{i=0}^{M-1}$ possess reflective symmetry: $\forall i \in \mathcal{I}_M$, $\text{Re}(a_i) = \text{Re}(a_{M-i})$, $\text{Im}(a_i) + \text{Im}(a_{M-i}) = 0$, where Re and Im represent the real and imaginary part of a complex number. In addition, since T is the smallest period by definition, we must have $a_1 = \overline{a_{M-1}} \neq 0$. Further, since $\mathbf{F}$ is smooth, the flow $\phi_t(\mathbf{x}_0) = \mathbf{x}(t)$ is also smooth in t (*Nijmeijer and Van der Schaft*, 1990). Thus, the error in the Fourier interpolation is uniformly bounded by twice the sum of the absolute value of truncated Fourier coefficients (*Boyd*, 2001). This leads to the uniform convergence

$$\lim_{M \longrightarrow \infty} |x(t) - S_M(t)| = 0. \tag{3.7}$$

Hence, one can easily approximate the original periodic trajectory uniformly to the desired level of accuracy by increasing M above a certain threshold.

It should be emphasized that the following theoretical sections of this chapter is purely concerned with deterministic linear models and noise free data.

3.3.1.2 Formulations of linear models

Given discrete M uniform samples of a scalar trajectory: $\{x_k\}_{k=0}^{M-1}$, consider constructing L-time delays of $x(t)$, $L \in \mathbb{N}$. Note that $L = 0$ corresponds to *no delays* considered. To avoid negative indices, we utilize the modulo operation defined in Equation (3.8),

$$\forall q \in \mathbb{N},\, \mathcal{P}(q) \triangleq q \pmod{M} = \begin{cases} q, & \text{if } q \in \mathcal{I}_M, \\ q - M\left\lfloor q/M \right\rfloor, & \text{otherwise} \end{cases} \tag{3.8}$$

to construct the L time-delay vector $\mathbf{Y}_k$,

$$\mathbf{Y}_k = \begin{bmatrix} x_{\mathcal{P}(k)} \\ x_{\mathcal{P}(k-1)} \\ \vdots \\ x_{\mathcal{P}(k-L)} \end{bmatrix} \in \mathbb{R}^{L+1}, \tag{3.9}$$

where $k \in \mathcal{I}_M$, $\lfloor \cdot \rfloor$ is the floor function. Considering Fourier interpolation, we have

$$\forall q \in \mathcal{I}_M, \quad x_{\mathcal{P}(q)} = \sum_{i \in \mathcal{I}_M} a_i \omega^{qi}, \quad \omega \triangleq e^{-j\frac{2\pi}{M}} \in \mathbb{C}, \tag{3.10}$$

which is also true for $q \notin \mathcal{I}_M$

$$\begin{aligned} x_{\mathcal{P}(q)} &= S_M((q - M\left\lfloor q/M \right\rfloor)\Delta t) = \sum_{i \in \mathcal{I}_M} a_i e^{-j\frac{2\pi i(q - M\lfloor q/M \rfloor)}{M}} \\ &= \sum_{i \in \mathcal{I}_M} a_i \omega^{qi}. \end{aligned} \tag{3.11}$$

Using Equation (3.4), we can rewrite the L time-delay vector $\mathbf{Y}_k$ in Equation (3.9) in the Fourier basis as

$$\mathbf{Y}_k = \mathbf{\Omega}_{k,L}\mathbf{a}, \tag{3.12}$$

where $\forall k \in \mathcal{I}_M$, $\mathbf{\Omega}_{k,L} \triangleq \begin{bmatrix} 1 & \omega^{k} & \omega^{2k} & \dots & \omega^{(M-1)k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{k-L} & \omega^{2(k-L)} & \dots & \omega^{(M-1)(k-L)} \end{bmatrix}$, $\mathbf{a} \triangleq \begin{bmatrix} a_0 \\ \vdots \\ a_{M-1} \end{bmatrix} \in \mathbb{C}^{M\times 1}$.

The problem of finding the minimal time delay required for the linear model with L time delays in Equation (2.28) to *perfectly* predict the data $\{x_k\}_{k=0}^{M-1}$ is equivalent to the existence of the *delay transition matrix* $\mathbf{K}$ such that,

$$x_{\mathcal{P}(k+1)} = \mathbf{K}^\top \mathbf{Y}_k, \quad \forall k \in \mathcal{I}_M, \tag{3.13}$$

where

$$\mathbf{K} = \begin{bmatrix} K_0 & K_1 & \dots & K_L \end{bmatrix}^\top \in \mathbb{R}^{(L+1)\times 1},$$

and

$$x_{\mathcal{P}(k+1)} = \mathbf{\Upsilon}_k^\top \mathbf{a}, \tag{3.14}$$

where

$$\mathbf{\Upsilon}_k \triangleq \begin{bmatrix} 1 & \omega^{k+1} & \omega^{2(k+1)} & \dots & \omega^{(M-1)(k+1)} \end{bmatrix}^\top. \tag{3.15}$$

For convenience, we vertically stack Equation (3.13) $\forall k \in \mathcal{I}_M$,

$$\mathbf{Y}_M \mathbf{K} = \mathbf{x}_M, \tag{3.16}$$

where $\mathbf{Y}_M \triangleq \begin{bmatrix} \mathbf{Y}_0^\top \\ \mathbf{Y}_1^\top \\ \vdots \\ \mathbf{Y}_{M-2}^\top \\ \mathbf{Y}_{M-1}^\top \end{bmatrix}$, $\mathbf{x}_M \triangleq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{M-1} \\ x_0 \end{bmatrix}$.

Based on the above formulation, we will discuss the minimal number of required time delays, the exact solution of $\mathbf{K}$ and the number of samples required on the time

domain in the following sections.

3.3.1.3 Minimal time delays

Recall that our goal is to determine the minimal number of time delays L, such that there exists a matrix $\mathbf{K}$ that satisfies the linear system Equation (3.13). Given one period of data, we can transform the system from the time domain to the spectral domain. Consider Equations (3.12) and (3.14), then Equation (3.16) is equivalent to the following, $\forall k \in \mathcal{I}_M$:

$$\mathbf{a}^\top \left(\begin{bmatrix} 1 \\ \omega^{k+1} \\ \omega^{2(k+1)} \\ \vdots \\ \omega^{(k+1)(M-1)} \end{bmatrix} - \begin{bmatrix} 1 & \dots & 1 \\ \omega^{k} & \dots & \omega^{k-L} \\ \omega^{2k} & \dots & \omega^{2(k-L)} \\ \vdots & \ddots & \vdots \\ \omega^{(M-1)k} & \dots & \omega^{(M-1)(k-L)} \end{bmatrix} \mathbf{K} \right) = 0. \tag{3.17}$$

This can be written as

$$\mathbf{a}^\top \left(\begin{bmatrix} 1 & & & & \\ & \omega & & & \\ & & \omega^2 & & \\ & & & \ddots & \\ & & & & \omega^{(M-1)} \end{bmatrix}^k \left(\begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ \vdots \\ \omega^{M-1} \end{bmatrix} - \begin{bmatrix} 1 & \dots & 1 \\ 1 & \dots & \omega^{-L} \\ 1 & \dots & \omega^{2(-L)} \\ \vdots & \ddots & \vdots \\ 1 & \dots & \omega^{(M-1)(-L)} \end{bmatrix} \mathbf{K} \right) \right) = 0. \tag{3.18}$$

We define the residual matrix $\mathbf{R}$ as,

$$\mathbf{R} \triangleq \begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ \vdots \\ \omega^{M-1} \end{bmatrix} - \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega^{-1} & \dots & \omega^{-L} \\ 1 & \omega^{-2} & \dots & \omega^{2(-L)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(M-1)} & \dots & \omega^{(M-1)(-L)} \end{bmatrix} \mathbf{K}. \tag{3.19}$$

Given one period of data, we vertically stack the above equation for each $k \in \mathcal{I}_M$. Recognizing the non-singular nature of a Vandermonde square matrix with distinct nodes, we have

$$\begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_{M-1} \\ a_0 & \omega a_1 & \omega^2 a_2 & \dots & \omega^{M-1} a_{M-1} \\ a_0 & \omega^2 a_1 & \omega^4 a_2 & \dots & \omega^{2(M-1)} a_{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_0 & \omega^{M-1} a_1 & \omega^{2(M-1)} a_2 & \dots & \omega^{(M-1)(M-1)} a_{M-1} \end{bmatrix} \mathbf{R} = \mathbf{0}. \tag{3.20}$$

This gives

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega & \dots & \omega^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{M-1} & \dots & \omega^{(M-1)(M-1)} \end{bmatrix} \begin{bmatrix} a_0 & & & \\ & a_1 & & \\ & & \ddots & \\ & & & a_{M-1} \end{bmatrix} \mathbf{R} = \mathbf{0}, \tag{3.21}$$

and thus

$$\begin{bmatrix} a_0 & & & \\ & a_1 & & \\ & & \ddots & \\ & & & a_{M-1} \end{bmatrix} \mathbf{R} = \mathbf{0}. \tag{3.22}$$

Note the equivalence between Equation (3.22) and Equation (3.16). Now, we consider the case when the Fourier spectrum is sparse with P non-zero coefficients, $P \in \mathbb{N}$ and $P \leqslant M$. Moreover, it is consistent with the finite point spectral resolution of Koopman operator appears in the laminar unsteady flows (*Mezić*, 2013). Denote the set of wave numbers associated with non-zero coefficients as,

$$\mathcal{I}_M^P \triangleq \{a_i \neq 0 | i \in \mathcal{I}_M\} = \{i_p\}_{p=0}^{P-1}, \tag{3.23}$$

with ascending order $0 \leqslant i_0 < i_1 < \ldots < i_{P-1} \leqslant M-1$, where $|\hat{\mathcal{I}}_M^P| = P \in \mathbb{N}$. Note that there is a reflective symmetry restriction on the Fourier spectrum.

The feasibility of using the number of time delays L to ensure the existence of a *real* solution $\mathbf{K}$ for the linear system is equivalent to the existence of the linear system $\mathbf{R} = \mathbf{0}$ after removing the rows that correspond to zero Fourier modes in $\mathbf{R}$, denoted as $\mathbf{R}_{\mathcal{I}_M^P}$,

$$\mathbf{R}_{\mathcal{I}_M^P} = \mathbf{0} \iff \mathbf{A}_{\mathcal{I}_M^P, L}\mathbf{K} = \mathbf{b}_{\mathcal{I}_M^P}, \tag{3.24}$$

where

$$\mathbf{A}_{\mathcal{I}_M^P, L} = \begin{bmatrix} 1 & \omega^{-i_0} & \ldots & \omega^{-Li_0} \\ 1 & \omega^{-i_1} & \ldots & \omega^{-Li_1} \\ 1 & \omega^{-i_2} & \ldots & \omega^{-Li_2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-i_{P-1}} & \ldots & \omega^{-Li_{P-1}} \end{bmatrix} \in \mathbb{C}^{P \times (L+1)}, \tag{3.25}$$

and

$$\mathbf{b}_{\mathcal{I}_M^P} = \begin{bmatrix} \omega^{i_0} \\ \omega^{i_1} \\ \omega^{i_2} \\ \vdots \\ \omega^{i_{P-1}} \end{bmatrix} \in \mathbb{C}^{P \times 1}. \tag{3.26}$$

Before presenting the main theorem Theorem III.5, we define the Vandermonde matrix in Definition III.2 and introduce Lemma III.3 and Lemma III.4.

Definition III.2. Vandermonde matrix with nodes as $\alpha_0, \alpha_1, \ldots, \alpha_{M-1} \in \mathbb{C}$ of order N is defined as,

$$\mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1}) \triangleq \begin{bmatrix} 1 & \alpha_0 & \ldots & \alpha_0^{N-1} \\ 1 & \alpha_1 & \ldots & \alpha_1^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_{M-1} & \ldots & \alpha_{M-1}^{N-1} \end{bmatrix}.$$

Lemma III.3. *$\forall M, N \in \mathbb{N}$, the Vandermonde matrix $\mathbf{A} = \mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ constructed from distinct $\{\alpha_i\}_{i \in \mathcal{I}_M}, \alpha_i \in \mathbb{C}$, has the two properties,*

1. $\operatorname{rank}(\mathbf{A}) = \min(M, N)$,
2. *if $\mathbf{A}$ has full column rank, $\forall Q \in \mathbb{N}, Q \leqslant M$, the rank of the submatrix $\mathbf{A}'$ by arbitrarily selecting Q rows is* $\min(Q, N)$.

Proof.

$$\mathbf{A} = \mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1}) = \begin{bmatrix} 1 & \alpha_0 & \ldots & \alpha_0^{N-1} \\ 1 & \alpha_1 & \ldots & \alpha_1^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_{M-1} & \ldots & \alpha_{M-1}^{N-1} \end{bmatrix} \tag{3.27}$$

If $M \geqslant N$, then

$$\mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1}) = \begin{bmatrix} \mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{N-1}) \\ \mathbf{V}_N(\alpha_N, \ldots, \alpha_{M-1}) \end{bmatrix} \tag{3.28}$$

Since $\{\alpha_i\}_{i \in \mathcal{I}_M}$ are distinct, $\mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{N-1})$ is full rank with rank N. Since $M \geqslant N$, the row space of $\mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ and is fully spanned by the first N

rows, and is thus full rank. Likewise, if $M < N$,

$$\mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1}) = \begin{bmatrix} \mathbf{V}_M(\alpha_0, \alpha_1, \ldots, \alpha_{M-1}) & * \end{bmatrix} \tag{3.29}$$

Similarly, the first M columns are full rank and $\mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ is also full rank. Thus in either case, $\mathbf{V}_N(\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ is full rank with rank as $\min(M, N)$. To show the the second property, one can simply replace $\{\alpha_i\}_{i \in \mathcal{I}_M}$ with $\{\alpha_i\}_{i \in \mathcal{J}}$ in the above arguments. Since $|\mathcal{J}| = Q$, $\text{rank}\left(\mathbf{V}_N(\{\alpha_i\}_{i \in \mathcal{J}})\right) = \min(Q, N)$. □

Lemma III.4. *$\forall m, n \in \mathbb{N}, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^{m \times 1}$, $\exists \mathbf{x} \in \mathbb{C}^{n \times 1}$ s.t. $\mathbf{Ax} = \mathbf{b} \iff \exists \mathbf{x}' \in \mathbb{R}^{n \times 1}$ s.t. $\mathbf{Ax}' = \mathbf{b}$. Further, when the solution is unique, the above still holds and the solution is real.*

Proof. First, let's prove from left to right. If $\exists \mathbf{x} \in \mathbb{C}^{n \times 1}$, we have $\mathbf{Ax} = \mathbf{b}$. Note that $\overline{\mathbf{Ax}} = \bar{\mathbf{A}}\bar{\mathbf{x}} = \mathbf{A}\bar{\mathbf{x}} = \bar{\mathbf{b}} = \mathbf{b}$ then consider $\mathbf{x}' = \frac{\bar{\mathbf{x}} + \mathbf{x}}{2} \in \mathbb{R}^{n \times 1}$. $\mathbf{Ax}' = (\mathbf{Ax} + \mathbf{A}\bar{\mathbf{x}})/2 = (\mathbf{b} + \mathbf{b})/2 = \mathbf{b}$. Second, it is easy to show from right to left. Third, when uniqueness is added, note that $\mathbf{Ax} = \mathbf{b} \iff \mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$, it is easy to show both directions since it is impossible to have complex solution being unique and not real. □

Theorem III.5. *For a uniform sampling of $S_M(t)$ with length M and P non-zero coefficients in the Fourier spectrum, the minimal number of time delays L for a perfect prediction, i.e., one that satisfies Equation* (3.16) *is $P-1$. Moreover, when $L = P-1$, the solution is unique.*

Proof. Consider the discrete Fourier spectrum of $S_M(t)$ with M uniform samples per period. The perfect prediction using a time-delayed linear model requires the existence of a real $\mathbf{K}$ that satisfies Equation (3.16), which is equivalent to Equation (3.22). Therefore, Equation (3.16) and Equation (3.22) share the same solutions in $\mathbb{C}^{(L+1) \times 1}$. Since the Fourier spectrum contains only P non-zero coefficients, Equation (3.22) is equivalent to Equation (3.24). The necessary and sufficient condition to have a solution (not necessarily real) $\mathbf{K}$ for Equation (3.24) follows from the Rouché-Capelli

theorem *Meyer* (2000),

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{A}_{\mathcal{I}_M^P,L} & \mathbf{b}_{\mathcal{I}_M^P}\end{bmatrix}\right) = \operatorname{rank}\left(\mathbf{A}_{\mathcal{I}_M^P,L}\right). \tag{3.30}$$

Using the first property in Lemma III.3, $\operatorname{rank}(\mathbf{A}_{\mathcal{I}_M^P,L}) = \min(P, L+1)$. While for the augmented matrix,

$$\begin{aligned}
&\operatorname{rank}\left(\begin{bmatrix}\mathbf{A}_{\mathcal{I}_M^P,L} & \mathbf{b}_{\mathcal{I}_M^P}\end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{b}_{\mathcal{I}_M^P} & \mathbf{A}_{\mathcal{I}_M^P,L}\end{bmatrix}\right) \qquad (3.31)\\
&= \operatorname{rank}\left(\begin{bmatrix}
\omega^{i_0} & 1 & \omega^{-i_0} & \dots & \omega^{-Li_0}\\
\omega^{i_1} & 1 & \omega^{-i_1} & \dots & \omega^{-Li_1}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
\omega^{i_{P-1}} & 1 & \omega^{-i_{P-1}} & \dots & \omega^{-Li_{P-1}}
\end{bmatrix}\right)\\
&= \operatorname{rank}\left(\begin{bmatrix}
\omega^{i_0} & & & \\
& \omega^{i_1} & & \\
& & \ddots & \\
& & & \omega^{i_{P-1}}
\end{bmatrix}\begin{bmatrix}
1 & \omega^{-i_0} & \dots & \omega^{-(L+1)i_0}\\
1 & \omega^{-i_1} & \dots & \omega^{-(L+1)i_1}\\
\vdots & \vdots & \ddots & \vdots\\
1 & \omega^{-i_{P-1}} & \dots & \omega^{-(L+1)i_{P-1}}
\end{bmatrix}\right)\\
&= \operatorname{rank}\left(\operatorname{diag}(\omega^{i_0},\dots,\omega^{i_{P-1}})\mathbf{V}_{L+2}(\omega^{-i_0},\dots,\omega^{-i_{P-1}})\right)\\
&= \operatorname{rank}\left(\mathbf{V}_{L+2}(\omega^{-i_0},\dots,\omega^{-i_{P-1}})\right)\\
&= \min(P, L+2).
\end{aligned}$$

Therefore, if $L+2 \leqslant P$, i.e., $L \leqslant P-2$, $\min(P, L+2) = L+2 \neq L+1 = \min(P, L+1)$. If $L+1 \geqslant P$, i.e., $L \geqslant P-1$, then $\min(P, L+2) = P = \min(P, L+1)$. So the minimal L for Equation (3.30) to hold is $P-1$, which makes $\mathbf{A}_{\mathcal{I}_M^P,L}$ an invertible Vandermonde square matrix. Thus the solution is unique in $\mathbb{C}^{(L+1)\times 1}$. From Lemma III.4, consider Equation (3.16), the solution is real. □

From the above Theorem III.5, we can easily derive Propositions 1 and 2 that are intuitive.

Proposition 1. *If there is only one frequency in the Fourier spectrum of $S_M(t)$, simply one time delay in the linear model is enough to perfectly recover the signal.*

Proposition 2. *If the Fourier spectrum of $S_M(t)$ is dense, then the maximum number of time delays, i.e., over the whole period $M-1$ is necessary to perfectly recover the signal.*

In retrospect, the result of the minimal number of time delays for a scalar time series is rather intuitive: any scalar signal with R frequencies corresponds to a certain observable of a $2R$-dimensional linear system. Since more time delays in linear model increases the number of eigenvalues in the corresponding linear system, one requires a minimum of $L = 2R - 1 = P - 1$ to match the number of eigenvalues.

3.3.1.4 Exact explicit solution for the linear model

One of the benefits of projecting the trajectory on a Fourier basis instead of a tailored basis such as proper orthogonal decomposition basis is the possibility of explicit theoretical analysis.

Two interesting facts have to be brought to the fore:

1. From Equation (3.24), it is clear that $\mathbf{K}$ is independent of the *quantitative value* of the Fourier coefficients, but only depends on the *pattern in the Fourier spectrum.*

2. For $L = P - 1$, $\mathbf{A}_{\mathcal{I}_M^P, L}$ is an invertible Vandermonde matrix, which implies the uniqueness of the solution $\mathbf{K}$.

Consider the general explicit formula for the inverse of a Vandermonde matrix (*Petersen et al.*, 2008). Note that $\mathbf{A}_{\mathcal{I}_M^P, P-1} = \mathbf{V}_P(\omega^{-i_0}, \ldots, \omega^{-i_{P-1}})$.

Thus

$$\mathbf{A}^{-1}_{\mathcal{I}^P_M,P-1} = \mathbf{V}_P^{-1}(\omega^{-i_0},\ldots,\omega^{-i_{P-1}}). \tag{3.32}$$

$$\mathbf{V}_P^{-1}(\omega^{-i_0},\ldots,\omega^{-i_{P-1}})_{mn} = (-1)^{m+1}\frac{\sum\limits_{\substack{0\leqslant k_1<\ldots<k_{P-m}\leqslant P-1\\ k_1,\ldots,k_{P-m}\neq n-1}} \omega^{-(i_{k_1}+\ldots+i_{k_{P-m}})}}{\prod\limits_{0\leqslant l\leqslant P-1, l\neq n-1}\omega^{-i_l}-\omega^{-i_{n-1}}}.$$

$$\begin{aligned}\mathbf{K}_m &= \mathbf{V}_P^{-1}(\omega^{-i_0},\ldots,\omega^{-i_{P-1}})_{mn}\mathbf{b}_{\mathcal{I}^P_M,L,n} \\ &= \sum_{n=1}^{P}(-1)^{m+1}\frac{\sum\limits_{\substack{0\leqslant k_1<\ldots<k_{P-m}\leqslant P-1\\ k_1,\ldots,k_{P-m}\neq n-1}} \omega^{-(i_{k_1}+\ldots+i_{k_{P-m}})}}{\prod\limits_{0\leqslant l\leqslant P-1, l\neq n-1}\omega^{-i_l}-\omega^{-i_{n-1}}}\omega^{i_{n-1}} \\ &= \sum_{n=1}^{P}(-1)^{m+1}\frac{\sum\limits_{0\leqslant k_1<\ldots<k_{P-m}\leqslant P-1} e^{\frac{j2\pi(i_{k_1}+\ldots+i_{k_{P-m}})}{M}}}{\prod\limits_{0\leqslant l\leqslant P-1, l\neq n-1} e^{\frac{j2\pi i_l}{M}}-e^{\frac{j2\pi i_{n-1}}{M}}}.\end{aligned} \tag{3.33}$$

where $1 \leqslant m, n \leqslant P$ and $\mathbf{K}_m \equiv K_{m-1}$.

Despite the explicit form, the above expression is not useful in practice. Without loss of generality, considering P is even, the computational complexity at least grows as $\binom{P}{P/2}$. As an example, for a moderate system with 50 non-sparse modes, $\binom{50}{25} \approx 1.2\times 10^{14}$.

3.3.1.5 Minimal number of samples

Projection of the trajectory onto a Fourier basis implies that at least one period of training data has to be obtained to be able to construct a linear system that has a unique solution corresponding to $\mathbf{K}^*$. However, we will show that in the time domain, a full period of data is not necessary to determine the solution $\mathbf{K}^*$ if the Fourier spectrum is sparse.

Denote the number of non-zero Fourier coefficients as $P \in \mathbb{N}$, and its index set as $\mathcal{I}^P_M$ as before. Instead of having a full period of data, without loss of generality, we

consider L time delays and select the Q rows in Equation (3.16), for which the index is denoted as $0 \leqslant k_0 < \ldots < k_{Q-1} \leqslant M-1$, and $Q \in \mathbb{N}, L+Q \leqslant M$. Therefore, we have the following equation in the time domain,

$$\begin{bmatrix} \mathbf{Y}_{k_0}^\top \\ \mathbf{Y}_{k_1}^\top \\ \vdots \\ \mathbf{Y}_{k_{Q-2}}^\top \\ \mathbf{Y}_{k_{Q-1}}^\top \end{bmatrix} \mathbf{K} = \begin{bmatrix} x_{\mathcal{P}(k_0+1)} \\ x_{\mathcal{P}(k_1+1)} \\ \vdots \\ x_{\mathcal{P}(k_{Q-2}+1)} \\ x_{\mathcal{P}(k_{Q-1}+1)} \end{bmatrix}. \tag{3.34}$$

Consider a Fourier transform and recall Equation (3.18). Choosing k over $k_0, \ldots, k_{Q-1}$, the above equation can be equivalently rewritten as

$$\begin{bmatrix} a_0 & \omega^{k_0} a_1 & \omega^{2k_0} a_2 & \ldots & \omega^{(M-1)k_0} a_{M-1} \\ a_0 & \omega^{k_1} a_1 & \omega^{2k_1} a_2 & \ldots & \omega^{(M-1)k_1} a_{M-1} \\ a_0 & \omega^{k_2} a_1 & \omega^{2k_2} a_2 & \ldots & \omega^{(M-1)k_2} a_{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_0 & \omega^{k_{Q-1}} a_1 & \omega^{2k_{Q-1}} a_2 & \ldots & \omega^{(M-1)k_{Q-1}} a_{M-1} \end{bmatrix} \mathbf{R} = \mathbf{0}. \tag{3.35}$$

Recall that only P Fourier coefficients are non-zero, and thus the above equation that

constrains $\mathbf{K}$ equivalently becomes

$$\begin{bmatrix} a_{i_0} & \omega^{k_0} a_{i_1} & \omega^{2k_0} a_{i_2} & \dots & \omega^{(P-1)k_0} a_{i_{P-1}} \\ a_{i_0} & \omega^{k_1} a_{i_1} & \omega^{2k_1} a_{i_2} & \dots & \omega^{(P-1)k_1} a_{i_{P-1}} \\ a_{i_0} & \omega^{k_2} a_{i_1} & \omega^{2k_2} a_{i_2} & \dots & \omega^{(P-1)k_2} a_{i_{P-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i_0} & \omega^{k_{Q-1}} a_{i_1} & \omega^{2k_{Q-1}} a_{i_2} & \dots & \omega^{(P-1)k_{Q-1}} a_{i_{P-1}} \end{bmatrix} \mathbf{R}_{\mathcal{I}_M^P} = \mathbf{0} \tag{3.36}$$

$$\iff \begin{bmatrix} 1 & \omega^{k_0} & \omega^{2k_0} & \dots & \omega^{(P-1)k_0} \\ 1 & \omega^{k_1} & \omega^{2k_1} & \dots & \omega^{(P-1)k_1} \\ 1 & \omega^{k_2} & \omega^{2k_2} & \dots & \omega^{(P-1)k_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{k_{Q-1}} & \omega^{2k_{Q-1}} & \dots & \omega^{(P-1)k_{Q-1}} \end{bmatrix} \begin{bmatrix} a_{i_0} & & & & \\ & a_{i_1} & & & \\ & & a_{i_2} & & \\ & & & \ddots & \\ & & & & a_{i_{P-1}} \end{bmatrix} \mathbf{R}_{\mathcal{I}_M^P} = \mathbf{0}$$

$$\iff \mathbf{V}_P(\omega^{k_0}, \dots, \omega^{k_{Q-1}}) \operatorname{diag}(a_{i_0}, \dots, a_{i_{P-1}}) \mathbf{R}_{\mathcal{I}_M^P} = \mathbf{0}. \tag{3.37}$$

Since $\{\omega^{k_j}\}_{j=0}^{Q-1}$ are distinct from each other, from Lemma III.3, $\operatorname{rank}(\mathbf{V}_P(\omega^{k_0}, \dots, \omega^{k_{Q-1}})) = \min(P, Q)$. Therefore, if we choose to *have training data points no less than the number of non-zero Fourier coefficients*, i.e., $Q \geqslant P$, then $\mathbf{V}_P(\omega^{k_0}, \dots, \omega^{k_{Q-1}})$ is full rank, which leads to $\mathbf{R}_{\mathcal{I}_M^P} = \mathbf{0}$. Meanwhile, the solution $\mathbf{K}$ is uniquely determined given $L = P - 1$. Therefore, given $Q \geqslant P$,

$$\begin{bmatrix} \mathbf{Y}_{k_0}^\top \\ \mathbf{Y}_{k_1}^\top \\ \vdots \\ \mathbf{Y}_{k_{Q-2}}^\top \\ \mathbf{Y}_{k_{Q-1}}^\top \end{bmatrix} \mathbf{K} = \begin{bmatrix} x_{\mathcal{P}(k_0+1)} \\ x_{\mathcal{P}(k_1+1)} \\ \vdots \\ x_{\mathcal{P}(k_{Q-2}+1)} \\ x_{\mathcal{P}(k_{Q-1}+1)} \end{bmatrix} \iff \mathbf{R}_{\mathcal{I}_M^P} = \mathbf{0} \overset{L=P-1}{\iff} \mathbf{K} = \mathbf{K}^* \tag{3.38}$$

For the case with minimal number of data samples, i.e., $Q = P$, a natural choice is to construct P rows of the future state from the P-th to $2P - 1$-th rows in Equa-

tion (3.16). In the above setting, in order to construct the linear system in time domain that has the unique solution $\mathbf{K}^*$ of Equation (3.24), we only require access to the first $2P$ snapshots of data. The key observation is that when the signal is sparse, instead of constructing the classic unitary DFT matrix (Equation (3.21) to Equation (3.22)), a random choice of P rows will be sufficient to uniquely determine a real solution $\mathbf{K}^*$. It has to be mentioned, however, that randomly chosen data points might not be optimal. For example, in Equation (3.36), the particular choice of sampling (i.e. the choice of Q rows), will determine the condition number of the complex Vandermonde matrix $\mathbf{V}_P(\omega^{k_0}, \ldots, \omega^{k_{Q-1}})$. The necessary and sufficient condition for *perfect* conditioning of a Vandermonde matrix is when $\{\omega^{k_j}\}_{j=0}^{Q-1}$ are uniformly spread on the unit circle (*Berman and Feuer*, 2007).

At first glance, our work might appear to be in the same vein as compressed sensing (CS) (*Donoho*, 2006; *Candes and Tao*, 2006) where a complete signal is extracted from only a few measurements. However, it should be emphasized that CS requires *random* projections from the whole field to extract information about a broadband signal in each measurement, while we simply follow the setup in modeling dynamical systems where only *deterministic* and sequential point measurements are available, and limited to a certain time interval.

Moreover, the above instance of accurately recovering the dynamical system without using a full period of data on the attractor is also reported elsewhere, for instance in sparse polynomial regression for data-driven modeling of dynamical systems (*Champion et al.*, 2019). Indeed, this is one of the key ideas behind SINDy (*Brunton et al.*, 2016b): one can leverage the prior knowledge of the existence of a sparse representation (for instance, in a basis of monomials), such that sparse regression can significantly reduce the amount of data required with no loss of information.

3.3.2 Vector Time Series

3.3.2.1 Formulations of linear models

In this section, we extend the above analysis in section 3.3.1 to the case of a vector dynamical system. Assuming the state vector has n components, given the time series of l-th component, $\{x_k^{(l)}\}_{k=0}^{M-1}$, $l = 1, \ldots, n$, we have, $\forall k \in \mathcal{I}_M$

$$\tilde{x}_{\mathcal{P}(k+1)} = \begin{bmatrix} x_{\mathcal{P}(k+1)}^{(1)} \\ \vdots \\ x_{\mathcal{P}(k+1)}^{(n)} \end{bmatrix} \in \mathbb{R}^{n \times 1}, \tag{3.39}$$

where $k \in \mathcal{I}_M, \forall 1 \leqslant l \leqslant n, l \in \mathbb{N}, x_{\mathcal{P}(k)}^{(l)} \in \mathbb{R}$, $n \in \mathbb{N}$. Rewrite Equation (3.13) in a vector form:

$$\tilde{x}_{\mathcal{P}(k+1)} = \tilde{\mathbf{K}}^\top \tilde{\mathbf{Y}}_k, \quad \forall k \in \mathcal{I}_M, \tag{3.40}$$

where $\tilde{x}_{\mathcal{P}(k+1)} \in \mathbb{R}^n$, $\tilde{\mathbf{K}} \in \mathbb{R}^{n(L+1) \times n}$ and

$$\tilde{\mathbf{Y}}_k = \begin{bmatrix} \mathbf{Y}_k^{(1)} \\ \vdots \\ \mathbf{Y}_k^{(n)} \end{bmatrix} \in \mathbb{R}^{n(L+1) \times 1}, \tag{3.41}$$

where $\mathbf{Y}_k^{(l)}$ are the L time-delay embeddings defined in Equation (3.9) for the l-th component of the state. In the present thesis, we treat the time-delay uniformly across all components.

3.3.2.2 Minimal time delays

Following similar procedures (projecting on Fourier basis) as before in section 3.3.1, denoting the Fourier coefficient of l-th component as $\mathbf{a}^{(l)} \in \mathbb{C}^{M \times 1}$, the following lemma which is an analogy to Equation (3.22) in the scalar case.

Lemma III.6. *The necessary and sufficient condition for the existence of a real solution* $\tilde{\mathbf{K}}$ *in Equation* (3.40) *is equivalent to the existence of a solution for the following linear system:*

$$\begin{bmatrix} \mathrm{diag}(\mathbf{a}^{(1)}) & \dots & \mathrm{diag}(\mathbf{a}^{(n)}) \end{bmatrix} \left(\begin{bmatrix} \mathbf{b}_{\mathcal{I}_M^M} & & \\ & \ddots & \\ & & \mathbf{b}_{\mathcal{I}_M^M} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{\mathcal{I}_M^M,L} & & \\ & \ddots & \\ & & \mathbf{A}_{\mathcal{I}_M^M,L} \end{bmatrix} \tilde{\mathbf{K}} \right) = \mathbf{0}. \tag{3.42}$$

The existence of the above solution is equivalent to the following relationship,

$$\begin{aligned} &\mathrm{rank}\left(\begin{bmatrix} \mathrm{diag}(\mathbf{a}^{(1)})\mathbf{A}_{\mathcal{I}_M^M,L} & \dots & \mathrm{diag}(\mathbf{a}^{(n)})\mathbf{A}_{\mathcal{I}_M^M,L} \end{bmatrix} \right) \\ &= \mathrm{rank}\left(\begin{bmatrix} \mathrm{diag}(\mathbf{a}^{(1)})\mathbf{A}_{\mathcal{I}_M^M,L} & \dots & \mathrm{diag}(\mathbf{a}^{(n)})\mathbf{A}_{\mathcal{I}_M^M,L} & \mathrm{diag}(\mathbf{a}^{(1)})\mathbf{b}_{\mathcal{I}_M^M} & \dots & \mathrm{diag}(\mathbf{a}^{(n)})\mathbf{b}_{\mathcal{I}_M^M} \end{bmatrix} \right). \end{aligned} \tag{3.43}$$

Proof. Given the definitions in Equations (3.39) to (3.41), note Equation (3.12), we have

$$\tilde{\mathbf{Y}}_k = \begin{bmatrix} \mathbf{\Omega}_{k,L} & & \\ & \ddots & \\ & & \mathbf{\Omega}_{k,L} \end{bmatrix} \begin{bmatrix} \mathbf{a}^{(1)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix}. \tag{3.44}$$

Recall Equation (3.15), note that

$$\mathbf{\Upsilon}_k = \mathbf{\Lambda}^k \mathbf{b}_{\mathcal{I}_M^M}, \tag{3.45}$$

where $\mathbf{\Lambda} \triangleq \begin{bmatrix} 1 & & & \\ & \omega & & \\ & & \ddots & \\ & & & \omega^{(M-1)} \end{bmatrix}$.

Moreover, note that

$$\boldsymbol{\Omega}_{k,L}^{\top} = \boldsymbol{\Lambda}^k \mathbf{A}_{\mathcal{I}_M^M, L}. \tag{3.46}$$

We rewrite Equation (3.40) for a given k using Equation (3.14) for the left hand side and Equation (3.44) for the right hand side in Equation (3.40),

$$\begin{bmatrix} \boldsymbol{\Upsilon}_k^{\top} & & \\ & \ddots & \\ & & \boldsymbol{\Upsilon}_k^{\top} \end{bmatrix} \begin{bmatrix} \mathbf{a}^{(1)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix} = \tilde{\mathbf{K}}^{\top} \begin{bmatrix} \boldsymbol{\Omega}_{k,L} & & \\ & \ddots & \\ & & \boldsymbol{\Omega}_{k,L} \end{bmatrix} \begin{bmatrix} \mathbf{a}^{(1)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix}. \tag{3.47}$$

Using Equations (3.45) and (3.46) for the above, we have

$$\begin{bmatrix} \mathbf{a}^{(1)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix}^{\top} \left(\begin{bmatrix} \boldsymbol{\Upsilon}_k & & \\ & \ddots & \\ & & \boldsymbol{\Upsilon}_k \end{bmatrix} \right) - \begin{bmatrix} \boldsymbol{\Omega}_{k,L}^{\top} & & \\ & \ddots & \\ & & \boldsymbol{\Omega}_{k,L}^{\top} \end{bmatrix} \tilde{\mathbf{K}} = \mathbf{0}, \tag{3.48}$$

$$\begin{bmatrix} \mathbf{a}^{(1)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Lambda}^k & & \\ & \ddots & \\ & & \boldsymbol{\Lambda}^k \end{bmatrix} \Bigg(\begin{bmatrix} \mathbf{b}_{\mathcal{I}_M^M} & & \\ & \ddots & \\ & & \mathbf{b}_{\mathcal{I}_M^M} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{\mathcal{I}_M^M, L} & & \\ & \ddots & \\ & & \mathbf{A}_{\mathcal{I}_M^M, L} \end{bmatrix} \tilde{\mathbf{K}} \Bigg) = \mathbf{0}. \tag{3.49}$$

Considering $k = 0, 1, \ldots, M-1$, we stack $\begin{bmatrix} \mathbf{a}^{(1)} \\ \vdots \\ \mathbf{a}^{(n)} \end{bmatrix}^\top \begin{bmatrix} \mathbf{\Lambda}^k & & \\ & \ddots & \\ & & \mathbf{\Lambda}^k \end{bmatrix}$ row by row as

$$\begin{aligned}
&\begin{bmatrix} a_0^{(1)} & \ldots & a_{M-1}^{(1)} & \ldots & a_0^{(n)} & \ldots & a_{M-1}^{(n)} \\ a_0^{(1)} & \ldots & \omega^{M-1} a_{M-1}^{(1)} & \ldots & a_0^{(n)} & \ldots & \omega^{M-1} a_{M-1}^{(n)} \\ \vdots & \ddots & \vdots & \ldots & \vdots & \ddots & \vdots \\ a_0^{(1)} & \ldots & \omega^{(M-1)^2} a_{M-1}^{(1)} & \ldots & a_0^{(n)} & \ldots & \omega^{(M-1)^2} a_{M-1}^{(n)} \end{bmatrix} \\
&= \mathbf{V}_M(\{\omega^j\}_{j=0}^{M-1}) \begin{bmatrix} \mathbf{I} & \ldots & \mathbf{I} \end{bmatrix} \operatorname{diag}(\{\mathbf{a}^{(l)}\}_{l=1}^n) \\
&= \mathbf{V}_M(\{\omega^j\}_{j=0}^{M-1}) \begin{bmatrix} \operatorname{diag}(\mathbf{a}^{(1)}) & \ldots & \operatorname{diag}(\mathbf{a}^{(n)}) \end{bmatrix}. \qquad (3.50)
\end{aligned}$$

Then plug the above equality into Equation (3.49), and notice the non-singularity of $\mathbf{V}_M(\{\omega^j\}_{j=0}^{M-1})$, for $k = 0, 1, \ldots, M-1$, Equation (3.49) can be rewritten as

$$\begin{aligned}
&\begin{bmatrix} \operatorname{diag}(\mathbf{a}^{(1)}) & \ldots & \operatorname{diag}(\mathbf{a}^{(n)}) \end{bmatrix} \Bigg(\begin{bmatrix} \mathbf{b}_{\mathcal{I}_M^M} & & \\ & \ddots & \\ & & \mathbf{b}_{\mathcal{I}_M^M} \end{bmatrix} \\
&- \begin{bmatrix} \mathbf{A}_{\mathcal{I}_M^M, L} & & \\ & \ddots & \\ & & \mathbf{A}_{\mathcal{I}_M^M, L} \end{bmatrix} \tilde{\mathbf{K}} \Bigg) = \mathbf{0}. \qquad (3.51)
\end{aligned}$$

From the Rouché-Capelli theorem *Meyer* (2000), the necessary and sufficient con-

dition for the existence of a complex solution to Equation (3.51) is,

$$\begin{aligned}
&\operatorname{rank}\left(\begin{bmatrix}\operatorname{diag}(\mathbf{a}^{(1)})\mathbf{A}_{\mathcal{I}_M^M,L} & \dots & \operatorname{diag}(\mathbf{a}^{(n)})\mathbf{A}_{\mathcal{I}_M^M,L}\end{bmatrix}\right)\\
&= \operatorname{rank}\Big(\Big[\operatorname{diag}(\mathbf{a}^{(1)})\mathbf{A}_{\mathcal{I}_M^M,L}\dots\operatorname{diag}(\mathbf{a}^{(n)})\mathbf{A}_{\mathcal{I}_M^M,L} \qquad (3.52)\\
&\qquad\qquad \operatorname{diag}(\mathbf{a}^{(1)})\mathbf{b}_{\mathcal{I}_M^M}\dots\operatorname{diag}(\mathbf{a}^{(n)})\mathbf{b}_{\mathcal{I}_M^M}\Big]\Big). \qquad (3.53)
\end{aligned}$$

Note that since the above procedures are can be retained in Equation (3.40), Equation (3.40) and Equation (3.51) share the same solution in $\mathbb{C}^{n(L+1)\times n}$. From Lemma III.4, Equation (3.52) is also the necessary and sufficient condition for Equation (3.40) to have a real solution. □

Next, with the introduction of the Krylov subspace in Definition III.7 which frequently appears in the early literatures of DMD (*Rowley et al.*, 2009; *Schmid*, 2010), we present Remark 1 and Remark 2 from Equation (3.42) that interprets and reveals the possibility of using *less* embeddings than the corresponding sufficient condition for the scalar case in Theorem III.5.

Definition III.7 (Krylov subspace)**.** For $n, r \in \mathbb{N}$, $\mathbf{A} \in \mathbb{C}^{n\times n}$, $\mathbf{b} \in \mathbb{C}^{n\times 1}$, Krylov subspace is defined as

$$\mathcal{K}_r(\mathbf{A}, \mathbf{b}) = \operatorname{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{r-1}\mathbf{b}\}. \qquad (3.54)$$

Remark 1 (Geometric interpretation)**.** *For $j = 1, \dots, n$, define $\mathbf{c}^{(j)} \triangleq \operatorname{diag}(\mathbf{a}^{(j)})\mathbf{b}_{\mathcal{I}_M^M}$, and $\mathcal{E}_L^{(j)}$ as the column space of $\operatorname{diag}(\mathbf{a}^{(j)})\mathbf{A}_{\mathcal{I}_M^M,L}$. The existence of the solution in Equation (3.42) is then equivalent to*

$$\begin{aligned}
&\forall j \in \{1, \dots, n\}, \mathbf{c}^{(j)} \in \mathcal{W}_L \triangleq \mathcal{E}_L^{(1)} \oplus \dots \oplus \mathcal{E}_L^{(n)}\\
&\iff \operatorname{span}\{\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(n)}\} \subseteq \mathcal{W}_L, \qquad (3.55)
\end{aligned}$$

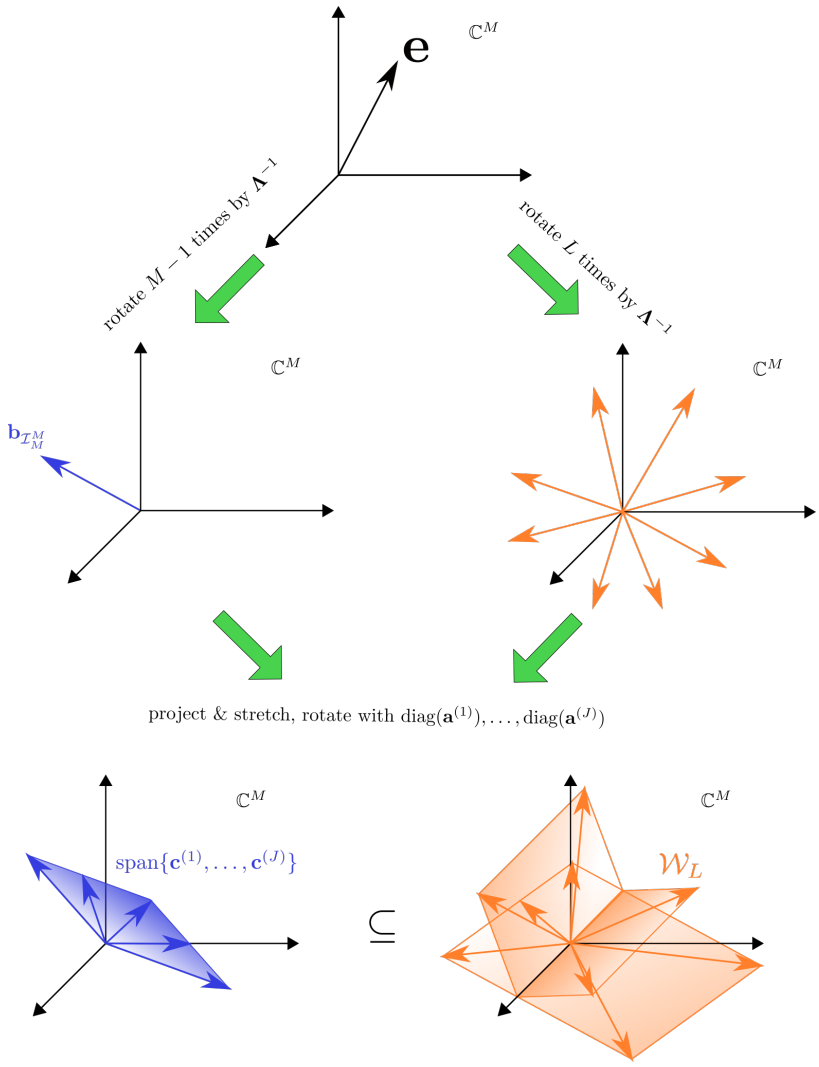


Figure 3.1: Illustration of the geometrical interpretation of Lemma III.6.

where $\mathcal{W}_L$ is the column space from all components, and $\oplus$ is the direct sum operation between vector spaces. Note that the column space of $\mathbf{A}_{\mathcal{I}_M^M,L}$ can represented as a Krylov subspace $\mathcal{K}_{L+1}(\mathbf{\Lambda}^{-1},\mathbf{e})$, where

$$\mathbf{e} \triangleq \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^\top, \tag{3.56}$$

$$\mathbf{\Lambda} \triangleq \operatorname{diag}(\omega^0,\dots,\omega^{M-1}). \tag{3.57}$$

A geometric interpretation of the above expressions is shown in Figure 3.1: for each j, $\mathbf{b}_{\mathcal{I}_M^M} = \mathbf{\Lambda}^{-(M-1)}\mathbf{e}$ and $\mathbf{e}$ are projected, stretched *and* rotated *using the j-th Fourier spectrum diagonal matrix* $\operatorname{diag}(\mathbf{a}^{(j)})$ *yields $\mathcal{E}_L^{(j)}$ and its total column subspace $\mathcal{W}_L$. If all of the projected and stretched $\mathbf{b}_M$'s are contained in $\mathcal{W}_L$, a real solution exists for Equation* (3.40)*. Notice that in Equation* (3.55)*, $\forall i \neq j$, $\mathcal{E}_L^{(i)}$ expands the column space $\mathcal{E}_L^{(j)}$ to include $\mathbf{c}^{(j)}$. Thus, the minimal number of time delays required in the vector case as in Equation* (3.40) *can be smaller than that of the scalar case.*

Remark 2 (Interplay between Fourier spectra)**.** *The vector case involves the interaction between the n different Fourier spectra corresponding to each component of the state. This complicates the derivation of an explicit result for the minimal number of time delays as in the scalar case (Theorem III.5). We note two important observations that illustrate the impact of the interplay between the n Fourier spectra:*

- *To ensure $\mathbf{c}^{(j)}$ lies in $\mathcal{W}_L$, each $\mathcal{E}_L^{(j)}$ should provide distinct vectors to maximize the dimension of $\mathcal{W}_L$. If a linear dependency is present in $\{\mathbf{a}^{(j)}\}_{j=1}^n$, Equation* (3.55) *no longer holds.*

- *Since $\mathbf{c}^{(j)}$ is projected using* $\operatorname{diag}(\mathbf{a}^{(j)})$*, if $\mathbf{a}^{(i)\top}\mathbf{a}^{(j)} = 0$, $\mathcal{E}_L^{(i)}$ will not contribute to increasing the dimension of $\mathcal{W}_L$.*

Drawing insight from the representation of the column space of $\mathbf{A}_{\mathcal{I}_M^M,L}$ as the Krylov subspace in Remark 1, we present a connection between the *output control-*

lability from linear system control theory (*Kreindler and Sarachik*, 1964), and the number of time delays required for linear models in a general sense.

Definition III.8 (Output controllability)**.** Consider a linear system with state vector $\mathbf{x}(t) \in \mathbb{C}^{M\times 1}$, $M \in \mathbb{N}$, $t \in \mathbb{R}^+$,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{3.58}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \tag{3.59}$$

where $\mathbf{A} \in \mathbb{C}^{M\times M}$, $\mathbf{B} \in \mathbb{C}^{M\times N}$, $\mathbf{C} \in \mathbb{C}^{P\times M}$, $\mathbf{D} \in \mathbb{C}^{P\times N}$. $\mathbf{y}(t) \in \mathbb{C}^{P\times 1}$ is the output vector. The above system is said to be output controllable if for any $\mathbf{y}(0), \mathbf{y}' \in \mathbb{C}^{P\times 1}$, there exists $t_1 \in \mathbb{R}^+, t_1 < +\infty$ and $\mathbf{u}' \in \mathbb{C}^{N\times 1}$, such that under such input and initial conditions, the output vector of the linear system can be transferred from $\mathbf{y}(0)$ to $\mathbf{y}' = \mathbf{y}(t_1)$.

Recall that the necessary and sufficient condition (*Kreindler and Sarachik*, 1964; *Gruyitch*, 2018) for a linear system to be output controllable is given in Definition III.9. A natural definition for the output controllability index that is similar to the controllability and observability index is given in Definition III.10. We summarize the conclusion in Theorem III.12 that the output controllability index minus one is a tight upper bound for the number of time delays required for the linear model in the general sense. We again emphasize that the particular linear system with input and output in Theorem III.12 is solely *induced* by the Fourier spectrum of the non-linear dynamical system on the attractor.

Definition III.9 (Output controllability test)**.** The system in Equations (3.58) and (3.59) is output controllable if and only if,

$$\mathcal{OC}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}; M) \triangleq \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \dots & \mathbf{CA}^{M-1}\mathbf{B} & \mathbf{D} \end{bmatrix}$$

is full rank. Note that when $\mathbf{D} = \mathbf{0}$, we omit $\mathbf{D}$ in the notation.

Definition III.10 (Output controllability index)**.** If the system in Equations (3.58) and (3.59) is output controllable, then the output controllability index is defined as the least integer μ such that $\mathcal{OC}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}; \mu) \in \mathbb{C}^{P \times (\mu+1)N}$ is full rank.

Lemma III.11. *For any matrix* $\mathbf{A}$ *that is a horizontal stack of diagonal matrices, the row elimination matrix* $\mathbf{E}$ *that removes any row that is a zero vector leads to a full rank matrix with the rank of original matrix. Moreover,* $\mathbf{E}^\top \mathbf{E}\mathbf{A} = \mathbf{A}$.

Proof. For $m, n \in \mathbb{N}$, consider n diagonal matrices in $\mathbf{A}$, for $j = 1, \ldots, n$, with the j-th diagonal matrices being $\text{diag}(\mathbf{a}^{(j)}) \in \mathbb{C}^{m \times m}$. $\mathbf{a}^{(j)} = \begin{bmatrix} \mathbf{a}_1^{(j)} & \mathbf{a}_2^{(j)} & \ldots \mathbf{a}_m^{(j)} \end{bmatrix}^\top$. Thus

$$\mathbf{A} = \begin{bmatrix} \text{diag}(\mathbf{a}^{(1)}) & \text{diag}(\mathbf{a}^{(2)}) & \ldots & \text{diag}(\mathbf{a}^{(m)}) \end{bmatrix} \in \mathbb{C}^{m \times mn}.$$

We define the following row index set that describes the row that is not a zero row vector in $\mathbf{A}$.

$$\Gamma = \{l | l \in \{1, \ldots, m\}, \exists j \in \{1, \ldots, m\}, \mathbf{a}_l^{(j)} \neq 0\}, \tag{3.60}$$

where we further order the index in Γ as

$$1 \leqslant \gamma_1 < \gamma_2 < \ldots < \gamma_P \leqslant m,$$

where $P = |\Gamma|$. Now we construct the row elimination matrix $\mathbf{E} \in \mathbb{C}^{P \times m}$ from Γ with

$$i \in \{1, \ldots, P\}, j \in \{1, \ldots, m\}, \mathbf{E}_{ij} = \delta_{\gamma_i, j}. \tag{3.61}$$

For $\mathbf{EA}$, since $\mathbf{E}$ only removes the zero row vector, the rank of the matrix $\mathbf{EA}$ is the same as $\mathbf{A}$. To show $\mathbf{EA}$ is full rank, simply consider the following procedure:

From the definition of Γ, on each row with row index $i = 1, \ldots, P$, there are non-zero entries. Start by choosing an entry, denoted as $\mathbf{a}_{\gamma_i}^{j_i}$ that is non-zero (while the choice of j_i is not unique). Then, one can simply perform column operations that switch the column with index j_i corresponding to the non-zero entry of i-th row, with the current i-th column. These operations can be iteratively performed, after which the following matrix is obtained:

$$\mathbf{EAR} = \begin{bmatrix} \mathbf{a}_{\gamma_1}^{j_1} & & & & * \\ & \mathbf{a}_{\gamma_2}^{j_2} & & & * \\ & & \ddots & & * \\ & & & \mathbf{a}_{\gamma_P}^{j_P} & * \end{bmatrix}, \tag{3.62}$$

where $\forall i = 1, \ldots, P, \mathbf{a}_{\gamma_i}^{j_i} \neq 0$ and $\mathbf{R}$ is the elementary column operation matrix. Thus $\mathbf{EAR}$ is full rank, and $\mathbf{EA}$ is full rank.

Define $\mathbf{F} = \mathbf{E}^\top$, i.e., $\mathbf{F}_{jk} = \delta_{\gamma_k, j}$. Thus

$$\begin{aligned} &i, j \in \{1, \ldots, m\}, \mathbf{G}_{ij} \triangleq \mathbf{F}_{ik}\mathbf{E}_{kj} = \delta_{\gamma_k, i}\delta_{\gamma_k, j} \\ &= \sum_{k=1}^{P} \delta_{\gamma_k, i}\delta_{\gamma_k, j} = \begin{cases} 1, \ i = j \in \Gamma, \\ 0, \ \text{otherwise.} \end{cases} \end{aligned} \tag{3.63}$$

Therefore, $\mathbf{G}$ is simply a diagonal matrix that keeps the row with index in Γ unchanged, but makes the row zero when the index is not in Γ. However, the row index that is not in Γ corresponds to a zero row vector, and thus $\mathbf{GA} = \mathbf{A}$, i.e., $\mathbf{E}^\top\mathbf{EA} = \mathbf{A}$. □

Theorem III.12. *Following definitions in Equations* (3.56) *and* (3.57)*, consider the*

following induced linear dynamical system with output controllability index μ:

$$\dot{\mathbf{Z}} = \mathbf{AZ} + \mathbf{Bu}$$

$$\mathbf{y} = \mathbf{CZ}$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{\Lambda}^{-1} & & \\ & \ddots & \\ & & \mathbf{\Lambda}^{-1} \end{bmatrix} \in \mathbb{C}^{Mn \times Mn},$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{e} & & \\ & \ddots & \\ & & \mathbf{e} \end{bmatrix} \in \mathbb{C}^{Mn \times n},$$

$$\mathbf{C}' = \begin{bmatrix} \operatorname{diag}(\mathbf{a}^{(1)}) & \dots & \operatorname{diag}(\mathbf{a}^{(n)}) \end{bmatrix} \in \mathbb{C}^{M \times nM},$$

$$\mathbf{C} = \mathbf{E}\mathbf{C}' \in \mathbb{C}^{P \times nM},$$

where P is the number of non-zero row vectors in $\mathbb{C}'$, and $\operatorname{rank}(\mathbf{C}) = \operatorname{rank}(\mathbf{C}') = P$ *as indicated by Lemma III.11. Then, $\mu - 1$ is a tight upper bound on the minimal number of time delays that ensures the existence of solution of Equation* (3.42)*, and thus a perfect reconstruction of the dynamics.*

Proof. Consider

$$\mathcal{OC}(\mathbf{A},\mathbf{B},\mathbf{C};\mu) = \mathbf{C}\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{\mu-1}\mathbf{B}\end{bmatrix} \tag{3.64}$$

$$= \mathbf{C}\begin{bmatrix}\mathbf{I} & \mathbf{A} & \dots & \mathbf{A}^{\mu-1}\end{bmatrix}\begin{bmatrix}\mathbf{B} & & \\ & \ddots & \\ & & \mathbf{B}\end{bmatrix}$$

$$= \mathbf{E}\mathbf{C}'\begin{bmatrix}\mathbf{I} & & & \mathbf{\Lambda}^{-(\mu-1)} & & \\ & \ddots & \dots & & \ddots & \\ & & \mathbf{I} & & & \mathbf{\Lambda}^{-(\mu-1)}\end{bmatrix}\begin{bmatrix}\mathbf{e} & & \\ & \ddots & \\ & & \mathbf{e}\end{bmatrix}$$

$$= \mathbf{E}\begin{bmatrix}\operatorname{diag}(\mathbf{a}^{(1)})\mathbf{e} & \dots & \operatorname{diag}(\mathbf{a}^{(n)})\mathbf{e} & \dots & \operatorname{diag}(\mathbf{a}^{(1)})\mathbf{\Lambda}^{-(\mu-1)}\mathbf{e} & \dots & \operatorname{diag}(\mathbf{a}^{(n)})\mathbf{\Lambda}^{-(\mu-1)}\mathbf{e}\end{bmatrix}.$$

Following Definition III.8, for any integer $i \geqslant \mu$, $\mathcal{OC}(\mathbf{A},\mathbf{B},\mathbf{C};i)$ is full rank. Thus, $\forall v \in \mathbb{C}^{P\times 1}$, v lies in the column space of $\mathcal{OC}(\mathbf{A},\mathbf{B},\mathbf{C};i)$. Therefore, $\mathbf{F}v$ should lie in the column space of $\mathbf{F}\mathcal{OC}(\mathbf{A},\mathbf{B},\mathbf{C};i)$. Noticing Lemma III.11 and Remark 1, we have

$$\mathbf{F}v \in \operatorname{Col}(\mathbf{F}\mathcal{OC}(\mathbf{A},\mathbf{B},\mathbf{C};i)) = \mathcal{W}_{i-1}. \tag{3.65}$$

Now, consider $\forall j = 1,\dots,n$, $v^{(j)} = \mathbf{E}\operatorname{diag}(\mathbf{a}^{(j)})\mathbf{b}_{\mathcal{I}_M^M} \in \mathcal{C}^{P\times 1}$, from the above, we have

$$\mathbf{F}v^{(j)} = \mathbf{F}\mathbf{E}\operatorname{diag}(\mathbf{a}^{(j)})\mathbf{b}_{\mathcal{I}_M^M} = \operatorname{diag}(\mathbf{a}^{(j)})\mathbf{b}_{\mathcal{I}_M^M} = \mathbf{c}^{(j)} \in \mathcal{W}_{i-1}. \tag{3.66}$$

Since the minimal i for $\mathcal{OC}(\mathbf{A},\mathbf{B},\mathbf{C};i)$ to be full rank is μ, the output observability index is μ. Correspondingly, when the number of time delays $L = \mu - 1$, a solution exists for Equation (3.42), which makes $\mu - 1$ an upper bound for the minimal time delay in Lemma III.6. Finally, to show that the bounds are tight, consider that when $n = 1$, Theorem III.12 reverts to Theorem III.5 where $\mu = P$, and thus $\mu - 1 = P - 1$ is essentially the minimal number of time delays required. □

3.4 Numerics

In this section, we start with a simple example and discuss practical numerical considerations.

3.4.1 Sanity check: 5-mode sine signal

First, an explicit time series consisting of five frequencies with a long period $T = 100$ is considered:

$$x(t) = 0.3\cos(\frac{2\pi t}{100}) + 0.5\sin(\frac{4\pi t}{100}) + 0.9\cos(\frac{8\pi t}{100}) + 1.6\sin(\frac{16\pi t}{100}) + 1.2\cos(\frac{24\pi t}{100}). \tag{3.67}$$

Such a signal may be realized, for instance, by observing the first component of a 10-dimensional linear dynamical system. The sampling rate is set at 1 per unit time, which is arbitrary and considered for convenience, and the signal is sampled for two periods from $n = 0$ to $n = 199$. Thus we have a discretely sampled time series of length 200 as $\{x_n\}_{n=0}^{199}$ with $x_n = x(t)|_{t=n}$. Only the first 20% of the original signal is used, which is 40% of a full period with around 20 to 30 data points sampled. The variation in the number of data points is due to the fact that we fix the use of first 20% of trajectory, and then reconstruct the signal with a different number of time delays. We solve the least squares problem in the time domain with the iterative least squares solver `scipy.linalg.lstsq` (*Jones et al.*, 2014) with lapack driver as `gelsd`, and cutoff for small singular values as 10^{-15}. The analysis in Theorem III.5 implies that one can avoid using the *full period* of data for exact prediction. Numerical results are presented in Figure 3.2 with number of time delays $L = 9$. These results show that time delayed DMD, unlike non-linear models such as neural networks, avoid the requirement of a *full period* of data when the dynamics is expressible by a set of sparse harmonics. From Theorem III.5, the 5-mode signal has $P = 10$ non-zero

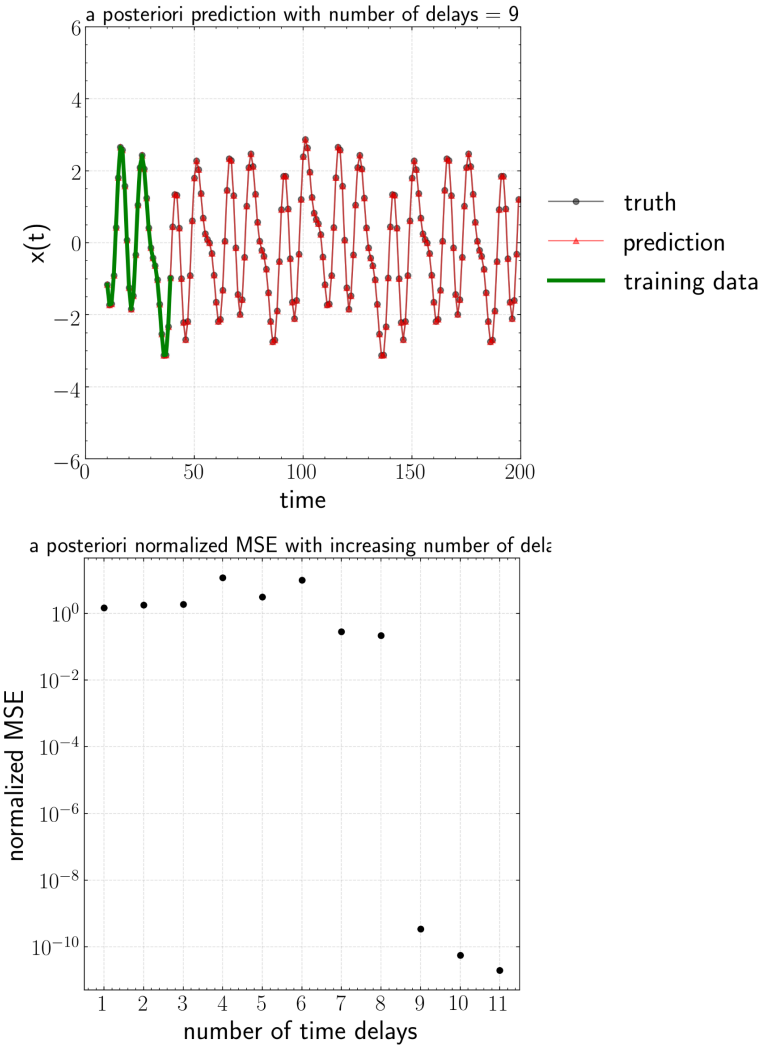


Figure 3.2: Top: A posteriori prediction vs ground truth, time delayed linear model with number of delays $L = 9$. Bottom: A posteriori MSE normalized by standard deviation of $x(t)$ vs number of time delays.

Fourier coefficients in the Fourier spectrum, and thus the least number of delays is $L = P - 1 = 9$, which agrees well with Figure 3.2 which shows the a posteriori mean square error normalized by the standard deviation of the data , between prediction and ground truth. Figure 3.2 clearly shows that a sharp decrease of a posteriori error when the number of delays $L = 9$.

Now we will consider a different scenario. As explained earlier, linear time delayed models can avoid the use of a *full period* of data if there is enough information to determine the solution within the first P states. Thus, if one increases the sampling rate, less data will be required to recover an accurate solution. However, one still needs to numerically compute the solution of a linear system, while the condition number grows with increasing sampling rates. As displayed in Figure 3.4, the condition number increases in both time and spectral domain formulations, with increasing sampling rate.

Using `scipy.linalg.lstsq` (*Jones et al.*, 2014) and a time domain formulation, we found that there is no visual difference between the truth and a posteriori prediction when the condition number is below 10^{13}, i.e., $M \leqslant 300$ in the spectral domain, or $M \leqslant 200$ in the time domain. However, as the condition number grows beyond 10^{13} (i.e. machine precision noise of even 10^{-16} can contaminate digits around 0.001), a posteriori prediction error can accumulate when $M = 400$ (Figure 3.3).

3.4.2 Temporal & Spectral Formulations

In practical terms, one can pursue two general formulations to numerically compute the delay transition matrix $\mathbf{K}$ in Equation (2.29):

1. *Formulation in time domain:* If all available delay vectors and corresponding future states are stacked, the direct solution of Equation (2.29) is a least square problem in the time domain with the requirement of at least P samples.

2. *Formulation in spectral domain:* In this approach, the Fourier signals from a

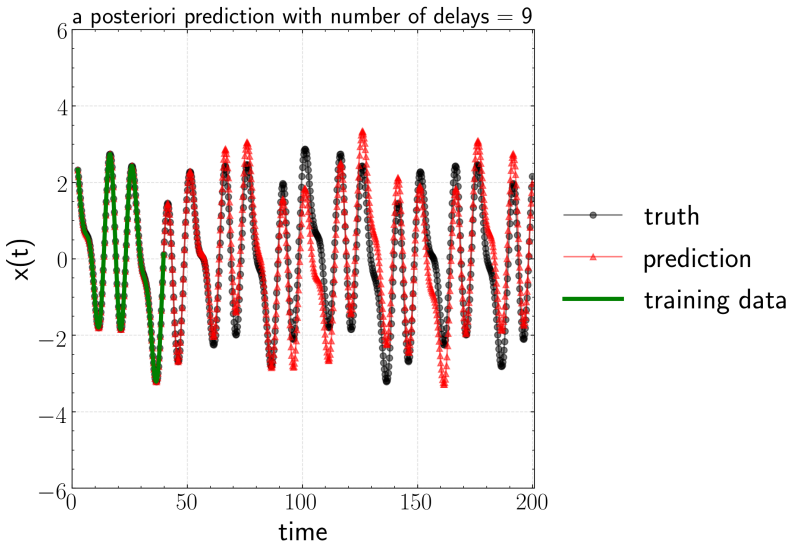


Figure 3.3: Prediction vs ground truth when sampling rate is excessive, e.g., $M = 400$ full period of data is extracted and Equation (3.24) is numerically solved.

3.4.3 Ill-conditioning due to excessive sampling rate

Consider signals that consist of a finite number of harmonics with the index set of Fourier coefficients as $\mathcal{I}_M^P$. Since the first half of the indices $i_0, \ldots, i_{P/2-1}$ is determined by the inherent period of each harmonic, these indices are independent of the number of samples per period M, as long as M satisfies the Nyquist condition. It is thus tempting to choose a relatively large sampling rate. However, this may not be favorable from a numerical standpoint. When $L = P - 1$ and the sampling rate is excessive compared to the potentially lower frequency dynamics of the system, each column could become nearly linearly dependent. We will now explore the circumstances under which the corresponding linear system in either the spectral or time domain can become ill-conditioned. It has to also be recognized that the denominator in Equation (3.33) consists of the difference between different nodes on the unit circle, and can therefore impact numerical accuracy.

The condition number of the Vandermonde matrix with complex nodes Equa-

tion (3.24) is also pertinent to the present discussion. It is well known that the condition number of a Vandermonde matrix grows exponentially with the order of matrix n when the nodes are real positive or symmetrically distributed with respect to the origin (*Córdova et al.*, 1990). When the nodes are complex, the numerical conditioning of a Vandermonde matrix can be as perfect as that of a DFT matrix, or as poor as that of the quasi-cyclic sequence (*Gautschi*, 1990). Specifically, it has been shown that a large square Vandermonde matrix is ill-conditioned unless its nodes are nearly uniformly spaced on or about the unit circle (*Pan*, 2016). Interestingly, for a rectangular Vandermonde matrix with n nodes and order N, i.e., $\mathbf{V}_N(z_1, \ldots, z_n)$, *Kunis and Nagel* (2018) provided a lower bound on the 2-norm condition number of the Vandermonde matrix that contains "nearly-colliding" nodes:

$$\kappa_2(\mathbf{V}_N(z_1, \ldots, z_n)) \geqslant \frac{\sqrt{6}}{\pi\tau} \approx \frac{0.77}{\tau}, \tag{3.68}$$

for all $\tau \leqslant 1$, i.e., "nearly colliding", where $\tau \triangleq N \min_{j \neq l} |t_j - t_l|_{\mathbb{T}}$, $|t_j - t_l|_{\mathbb{T}} \triangleq \min_{r \in \mathbb{Z}} |t_j - t_l + r|$. Applying the above result to Equation (3.24), when M is large enough so that $\tau \leqslant 1$ is satisfied[1], the lower bound of the 2-norm condition number will increase proportionally with the number of samples per period M. Thus, the tightly clustered nodes due to excessive sampling will lead to the ill-conditioning of the linear system in Equation (3.24).

3.4.4 Sub-sampling within Nyquist limits

Equation (3.68) shows that the tight clustering of nodes due to excessive sampling can lead to ill-conditioning. A straightforward fix would thus be to filter out unimportant harmonics, and re-sample the signal at a smaller sampling rate that can still capture the highest frequency retained in the filtering process. In this way, the nodes can be more favorably redistributed on the unit circle. Recall that, if the complex

[1] since $\tau = O(1/M)$

nodes of the Vandermonde matrix are uniformly distributed on a unit circle, then one arrives at a perfect conditioning of the Vandermonde matrix with condition number of one similar to the DFT matrix (*Pan*, 2016). Without any loss of generality, we assume the number of samples per period M is even. The wave numbers of sparse Fourier coefficients are denoted by $\mathcal{I}_M^P$. The sorted wave numbers are symmetrical with respect to $M/2$ and recall that the values of the first half of $\mathcal{I}_M^P$, i.e., $i_0, \ldots, i_{\frac{P}{2}-1}$ is independent of M, as long as the Nyquist condition is satisfied (*Landau*, 1967). Then, a continuous signal $x(t)$ is sub-sampled uniformly. Due to symmetry, the smallest number of samples per period M^* that preserves the signal is $2(i_{\frac{P}{2}-1}+1)$.

3.4.5 Effect of sampling rate, formulation domain, and numerical solver on model accuracy

To compare the impact of different solution techniques, we choose several off-the-shelf numerical methods to compute $\mathbf{K}$ in either the time domain or spectral domain. These methods include:

(i) `mldivide` from `MATLAB` (*MATLAB*, 2010), i.e., backslash operator which effectively uses QR/LU solver in our case;

(ii) `scipy.linalg.lstsq` (*Jones et al.*, 2014), which by default calls `gelsd` from `LAPACK` (*Anderson et al.*, 1999) to solve the minimum 2-norm least squares solution with SVD, and an algorithm based on divide and conquer;

(iii) Björck & Pereyra (BP) algorithm (*Björck and Pereyra*, 1970) which is designed to solve the Vandermonde system exactly in an efficient way exploiting the inherent structure. For a $n \times n$ matrix, instead of the standard Gaussian elimination with $O(n^3)$ arithmetic operations and $O(n^2)$ elements for storage, the BP algorithm only requires $n(n+1)(2O_M+3O_A)/2$[2] for arithmetic operations and no further storage than storing the roots and right hand side of the system.

[2] O_A and O_M denote addition/subtraction and multiplication/division.

As shown in Figure 3.4, the condition number increases exponentially with increasing number of samples per period M, leading to a significant deterioration of accuracy. Comparing the time and spectral domain formulations, Figure 3.4 shows that the solution for the spectral case is more accurate than the time domain solution when the sampling rate is low. This is not unexpected as one would need to perform FFT on a full period of data to find the appropriate Fourier coefficients in the spectral case. When $M > 600$, however, the spectral domain solutions obtained by BP and `mldivide` algorithms blow up, while the time domain solution is more robust in that the error is bounded. Note that the singular value decomposition - in `lstsq` and in `mldivide` that removes the components of the solution in the subspace spanned by less significant right singular vectors - is extremely sensitive to noise. Further, from Equation (3.36), the difference between the formulations in the spectral and time domains can be attributed to $\mathbf{V}_P(\omega^{k_0}, \ldots, \omega^{k_{Q-1}})$ and $\text{diag}(a_{i_0}, \ldots, a_{i_{P-1}})$, which could be ill-conditioned. Thus, regularization in the time domain formulation is more effective. Figure 3.4 also shows that, when the system becomes highly ill-conditioned, i.e., $M > 600$, `lstsq` with thresholding $\epsilon = 10^{-15}$ results in a more stable solution than `mldivide`.

It should be mentioned that the condition number computed in Figure 3.4 around the inverse of machine precision, i.e., $O(10^{16})$, should be viewed in a qualitative rather than quantitative sense (*Drmac et al.*, 2019).

3.4.6 Effect of the number of time delays L on condition number

By adding more time delays than the theoretical minimum, the dimension of the solution space grows, along with the features for least squares fitting. Accordingly, the null space becomes more dominant, and thus one should expect non-unique solutions with lower residuals. Note that, for simplicity, the following numerical analysis assumes the scalar case, i.e., $n = 1$.

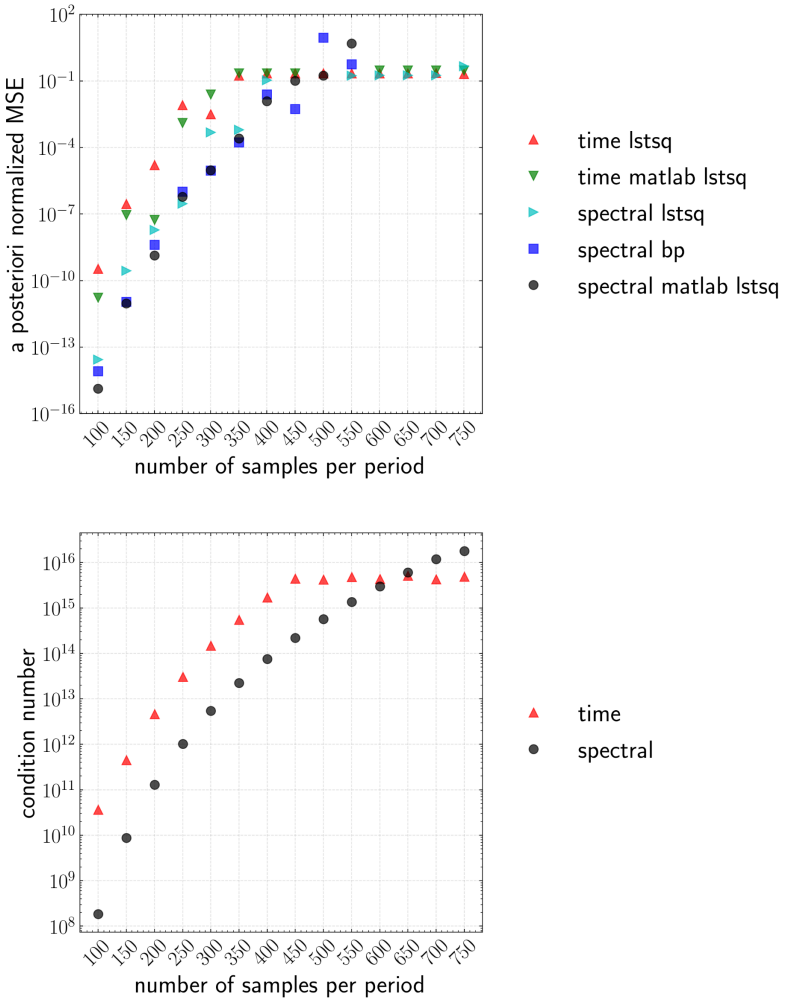


Figure 3.4: Top: A posteriori MSE normalized by the standard deviation of $x(t)$ with increasing sampling rate and different numerical solvers. Bottom: Numerical condition number with increasing sampling rate

For the complex Vandermonde system in Equation (3.24), following *Bazán* (2000), we discovered very distinct features of the asymptotic behavior of the solution, and the corresponding system in Equation (3.24) when the number of time delays $L \to \infty$.

(i) The norm of the minimum 2-norm solution of Equation (3.24) $\|\hat{\mathbf{K}}_L\|_2 \to 0$, as shown in Proposition 3.

(ii) An upper bound for the convergence rate of $\|\hat{\mathbf{K}}_L\|_2^2$ is derived in Lemma III.13.

(iii) An upper bound on the 2-norm condition number of Equation (3.24) is shown in Proposition 4 to scale with $1 + O(1/\sqrt{L})$.

Proposition 3. $\lim_{L\to\infty} \|\hat{\mathbf{K}}_L\|_2 = 0$, *where* $\hat{\mathbf{K}}_L$ *is the minimum 2-norm solution of Equation* (3.24).

Proof. To begin with, consider the following under-determined linear system for $f \in \mathbb{R}^N$, given $N \geqslant n$

$$\mathbf{V}_N(z_1, \ldots, z_n) f = \text{diag}(z_1, \ldots, z_n)\mathbf{e}, \tag{3.69}$$

where $\mathbf{e} = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^\top$. Denote f_N to be the minimum 2-norm solution. Suppose for all nodes, $i = 1, \ldots, n$, $|z_i| \leqslant 1$. Bazán *Bazán* (2000) showed that

$$\lim_{N\to+\infty} \|f_N\|_2 = 0. \tag{3.70}$$

Consider multiplying Equation (3.24) on both sides from the left with $\text{diag}(\omega^{Li_0}, \ldots, \omega^{Li_{P-1}})$. Notice that the diagonal matrix is non-singular for any $L \in \mathbb{N}$, and the inverse of per-

mutation matrix is its transpose. Then we have

$$\begin{bmatrix} \omega^{Li_0} & \omega^{(L-1)i_0} & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \omega^{Li_{P-1}} & \omega^{(L-1)i_{P-1}} & \dots & 1 \end{bmatrix} \mathbf{K} = \begin{bmatrix} \omega^{(L+1)i_0} \\ \vdots \\ \omega^{(L+1)i_{P-1}} \end{bmatrix}, \tag{3.71}$$

$$\begin{bmatrix} 1 & \omega^{i_0} & \dots & \omega^{Li_0} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{i_{P-1}} & \dots & \omega^{Li_{P-1}}, \end{bmatrix} \mathbf{P}^\top \mathbf{K} = \begin{bmatrix} \omega^{i_0} & & \\ & \ddots & \\ & & \omega^{i_{P-1}} \end{bmatrix}^{L+1} \mathbf{e}, \tag{3.72}$$

$$\mathbf{V}_{L+1}(\omega^{i_0}, \dots, \omega^{i_{P-1}}) f = (\mathrm{diag}(\omega^{i_0}, \dots, \omega^{i_{P-1}}))^{L+1} \mathbf{e}, \tag{3.73}$$

where $f \triangleq \mathbf{P}^\top \mathbf{K}$, $\mathbf{P} \in \mathbb{R}^{(L+1)\times(L+1)}$ is the column permutation matrix that reverses the column order in $\mathbf{A}_{\mathcal{I}_M^P, L}$. Note that a solution exists when $L+1 = P$ and it is not unique when $L+1 > P$. Denote f_L as the corresponding minimal 2-norm solution of Equation (3.73). From Equation (3.70), consider Equation (3.73) and take $L \to +\infty$, $\|f_L\|_2 \to 0$. The row permutation matrix does not change the 2-norm of a vector, and hence there is a one-to-one correspondence between the solution in Equation (3.73) and Equation (3.24), such that the corresponding minimal 2-norm solution for Equation (3.24) is $\hat{\mathbf{K}}_L \triangleq \mathbf{P} f_L$ thus $\|\hat{\mathbf{K}}_L\|_2 \to 0$. □

Lemma III.13. *$\forall L \geqslant P-1$, denote $\hat{\mathbf{K}}_L$ as the minimum 2-norm solution of Equation* (3.24)*. The following tight upper bound can be derived*

$$\|\hat{\mathbf{K}}_L\|_2^2 \leqslant \frac{\|\hat{\mathbf{K}}_{P-1}\|_2^2}{1 + \left\lfloor \frac{L-P+1}{M} \right\rfloor}. \tag{3.74}$$

Proof. For $q \in \mathbb{N}$, denote $L_q = qM + P - 1$. Note that in Equation (3.24), when $L = P-1$, the minimal 2-norm solution $\hat{\mathbf{K}}_{P-1}$ is also unique. Specifically we denote $\hat{\mathbf{K}}_{P-1} = \begin{bmatrix} \hat{K}_0 & \dots & \hat{K}_{P-1} \end{bmatrix}$. Note that, for any $L \geqslant P-1$, we can find $q = \left\lfloor \frac{L-P+1}{M} \right\rfloor$, such that $L \in \mathcal{T}_q \triangleq [L_q, L_{q+1})$. From the definition of the minimal 2-norm solution,

we have $\|\hat{\mathbf{K}}_L\|_2 \leqslant \|\hat{\mathbf{K}}_{L_q}\|_2$.

Consider $\mathbf{A}_{\mathcal{I}_M^P, L_q}$ and notice that for $q = 0$, i.e., $L_0 = P-1 \leqslant L < L_1 = M+P-1$, so $\|\hat{\mathbf{K}}_L\|_2 \leqslant \|\hat{\mathbf{K}}_{L_0}\|_2 = \|\hat{\mathbf{K}}_{P-1}\|_2$; for $q \geqslant 1$, for any $1 \leqslant j \leqslant P$, the j-th column of $\mathbf{A}_{\mathcal{I}_M^P, L_q}$ is duplicated with the $(j+kM)$-th column, $k = 1, \ldots, q$. For $q \geqslant 1$, $\mathbf{A}_{\mathcal{I}_M^P, L_q}$ in Equation (3.24), consider the following easily validated special class of real solutions,

$$\mathbf{K} = \begin{bmatrix} K_0 \\ \vdots \\ K_{P-1} \\ 0 \\ \vdots \\ 0 \\ K_M \\ \vdots \\ K_{L_1} \\ 0 \\ \vdots \\ 0 \\ \vdots \\ K_{qM} \\ \vdots \\ K_{L_q} \end{bmatrix}^\top \in \mathbb{R}^{1 \times (L_{q+1})}, \tag{3.75}$$

with the constraint that for any $1 \leqslant j \leqslant P$, $\sum_{l=0}^{q} K_{j-1+lM} = \hat{K}_{j-1}$. To find the minimal 2-norm solution, note that we have

$$\min \|\mathbf{K}\|_2^2 = \sum_{j=1}^{P} \min \sum_{l=0}^{q} K_{j-1+lM}^2. \tag{3.76}$$

From Jensen's inequality, $\forall j = 1, \ldots, P$,

$$\frac{\sum_{l=0}^{q} K_{j-1+lM}^2}{q+1} \geqslant \left(\frac{\sum_{l=0}^{q} K_{j-1+lM}}{q+1} \right)^2, \tag{3.77}$$

$$\sum_{l=0}^{q} K_{j-1+lM}^2 \geqslant \frac{\hat{K}_{j-1}^2}{q+1}, \tag{3.78}$$

where the equality holds when $K_{j-1+lM} = \hat{K}_{j-1}/(q+1)$ for $l = 0, \ldots, q$. Thus $\min\|\mathbf{K}\|_2^2 = \sum_{j=1}^{P} \hat{K}_{j-1}^2/(q+1) = \|\hat{\mathbf{K}}_{P-1}\|_2^2/(q+1)$. Since the above minimal norm is found within a special class of solutions in Equation (3.24), the general minimal 2-norm is

$$\|\hat{\mathbf{K}}_L\|_2^2 \leqslant \|\hat{\mathbf{K}}_{L_q}\|_2^2 \leqslant \|\hat{\mathbf{K}}_{P-1}\|_2^2/(q+1).$$

Combining both cases for $q = 0$ and $q \geqslant 1$, we have the desired result. □

Proposition 4. *Let P be the number of non-zero Fourier coefficients. $\forall L \geqslant P-1$, denote $\hat{\mathbf{K}}_{P-1}$ as the unique solution of Equation* (3.24)*. With the minimal number of time delays, the upper bound on the 2-norm condition number of the system is given by*

$$\begin{aligned} \kappa_2(\mathbf{A}_{\mathcal{I}_M^P, L}) &= \kappa_2(\mathbf{V}_{L+1}(\omega^{-i_0}, \ldots, \omega^{-i_{P-1}})) \\ &\leqslant 1 + \frac{d}{2}\left[1 + \sqrt{1 + \frac{4}{d}}\right], \end{aligned} \tag{3.79}$$

where

$$d \triangleq P\left[\left(1 + \frac{\|\hat{\mathbf{K}}_{P-1}\|_2^2}{(P-1)(1+\left\lfloor\frac{L-P+1}{M}\right\rfloor)\delta^2}\right)^{\frac{P-1}{2}} - 1\right], \tag{3.80}$$

$$\delta \triangleq \min_{0 \leqslant j < k \leqslant P-1} |\omega^{-i_j} - \omega^{-i_k}|. \tag{3.81}$$

Further, if $L \to \infty$, then $\kappa_2(\mathbf{A}_{\mathcal{I}_M^P, L}) \to 1$, i.e., perfect conditioning is achieved.

Proof. Consider the fact that the Vandermonde matrix $\mathbf{V}_N(z_1, \ldots, z_n)$ with n distinct

nodes $\{z_i\}_{i=1}^n$, $z_i \in \mathbb{C}$ of order N, $N \geqslant n$, i.e., $\mathbf{V}_N$ is full rank. The Frobenius-norm condition number is defined as $\kappa_F(\mathbf{V}_N) \triangleq \|\mathbf{V}_N\|_F \|\mathbf{V}_N^\dagger\|_F$, where $\dagger$ represents Moore-Penrose pseudoinverse. Bazán *Bazán* (2000) showed that if $\forall i = 1, \ldots, n$, with distinct $|z_i| \leqslant 1$, $N \geqslant n$, then

$$\kappa_F(\mathbf{V}_N) \leqslant n \left[1 + \frac{(n-1) + \|f_N\|_2^2 + \prod_{i=1}^n |z_i|^2 - \sum_{i=1}^n |z_i|^2}{(n-1)\delta^2} \right]^{\frac{n-1}{2}} \phi_N(\alpha, \beta), \quad (3.82)$$

where $\delta \triangleq \min\limits_{1 \leqslant i < j \leqslant n} |z_i - z_j|$, $\phi_N(\alpha, \beta) \triangleq \sqrt{\frac{1+\alpha^2+\ldots+\alpha^{2(N-1)}}{1+\beta^2+\ldots+\beta^{2(N-1)}}}$, $\alpha \triangleq \max\limits_{1 \leqslant j \leqslant n} |z_j|$, $\beta \triangleq \min\limits_{1 \leqslant j \leqslant n} |z_j|$.

The key to understand the behavior of the upper bound of $\kappa_2(\mathbf{V}_N)$, is to estimate the convergence rate of $\|f_N\|_2$ which is considered difficult for a general distribution of nodes *Bazán* (2000). For the particular case of Equation (3.24), we can show a tight upper bound in Lemma III.13. Thus, $\forall 1 \leqslant i \leqslant n, |z_i| = 1$, Equation (3.82) becomes,

$$\kappa_F(\mathbf{V}_N) \leqslant n \left(1 + \frac{\|f_N\|_2^2}{(n-1)\delta^2} \right)^{\frac{n-1}{2}}. \quad (3.83)$$

Now we note a general inequality between the condition number in the 2-norm and in the Frobenius norm *Bazán* (2000) by considering,

$$n - 2 < n - 2 + \kappa_2(\mathbf{V}_N) + \kappa_2^{-1}(\mathbf{V}_N) \leqslant \kappa_F(\mathbf{V}_N), \quad (3.84)$$

$$\kappa_2(\mathbf{V}_N) \leqslant \frac{1}{2} \left[\kappa_F(\mathbf{V}_N) - n + 2 + \sqrt{(\kappa_F(\mathbf{V}_N) - n + 2)^2 - 4} \right]. \quad (3.85)$$

The right hand side in Equation (3.85) is monotonically increasing with respect to $\kappa_F(\mathbf{V}_N)$. Therefore using the upper bound from Equation (3.83) in Equation (3.85), and some algebra we have the following upper bound, $\forall N > n$,

$$\kappa_2(\mathbf{V}_N) \leqslant 1 + \frac{d}{2} \left[1 + \sqrt{1 + \frac{4}{d}} \right], \quad (3.86)$$

where

$$d \triangleq n\left[\left(1+\frac{\|f_N\|_2^2}{(n-1)\delta^2}\right)^{\frac{n-1}{2}}-1\right]. \tag{3.87}$$

Finally, note that d monotonically increases with $\|f_N\|_2$, and thus with $n = P$, $N = L+1$, $z_l = \omega^{-i_l}$, $l = 0,\ldots,P-1$ and Lemma III.13, the desired upper bound is achieved. As $L \to \infty$, $\hat{\mathbf{K}}_L \to 0$ and $d \to 0$, and thus it is trivial to show that $\kappa_2(\mathbf{A}_{\mathcal{I}_M^P,L}) \to 1$. □

Remark 3. *Note that the bound in Proposition 4 does not demand a potentially restrictive condition on the number of time delays, i.e., $L+1 > 2(P-1)/\delta$ that is required in Bazán's work, which utilizes the Gershgorin circle theorem for the upper bound of the 2-norm condition number (Bazán, 2000). More recently, this constraint has been defined in the context of the nodes being "well-separated" (Kunis and Nagel, 2018). Applying such a result to our case, we have*

$$\kappa_2(\mathbf{A}_{\mathcal{I}_M^P,L}) \leqslant \sqrt{1+\frac{2}{\frac{\delta(L+1)}{2P-2}-1}} \tag{3.88}$$

since we have an estimation for the convergence rate of the minimal 2-norm solution. However, although our upper bound in Proposition 4 holds[3] for all $L \geqslant P-1$, it is too conservative compared to Bazán's upper bound when $L \to \infty$. To see this, denote $k_m \triangleq \min_{i,j\in\mathcal{I}_M^P, i,\neq j}\{|k| | k = (i-j) \bmod M\}$, i.e., the minimal absolute difference between any pair of distinct indices in $\mathcal{I}_M^P$, in the sense of modulo M. Assuming that the number of samples per period is large enough so that $M \gg 2\pi k_m$, we have $\delta = \sqrt{2\left[1-\cos(2\pi k_m/M)\right]} \approx 2\pi k_m/M = O(1/M)$. If we assume that the system with time delay L is far from being perfectly conditioned, we have $\kappa_F(\mathbf{V}_{L+1}) \gg P+2$,

[3] and is more general than Bazán's upper bound Equation (3.88)

which leads to the following approximation for our upper bound,

$$\begin{aligned}\kappa_2(\mathbf{V}_N) &\leqslant \frac{1}{2}\Big[\kappa_F(\mathbf{V}_{L+1}) - P + 2 \\ &+ \sqrt{(\kappa_F(\mathbf{V}_{L+1}) - P + 2)^2 - 4}\Big] \approx \kappa_F(\mathbf{V}_{L+1}) - P + 2 \\ &\leqslant d + 2. \end{aligned} \tag{3.89}$$

Hence, for an excessively sampled case, if L is small enough such that $\kappa_F(\mathbf{V}_{L+1}) \geqslant \kappa_2(\mathbf{V}_{L+1}) \gg P + 2$ holds but large enough such that

$$\frac{\|\hat{\mathbf{K}}_{P-1}\|_2^2}{(P-1)(1 + \lfloor \frac{L-P+1}{M} \rfloor)\delta^2} \ll 1, \tag{3.90}$$

then the approximated upper bound becomes

$$\begin{aligned} 2 + d &= 2 + P\left[\left(1 + \frac{\|\hat{\mathbf{K}}_{P-1}\|_2^2}{(P-1)(1 + \lfloor \frac{L-P+1}{M} \rfloor)\delta^2}\right)^{\frac{P-1}{2}} - 1\right], \\ &\approx 2 + \frac{P\|\hat{\mathbf{K}}_{P-1}\|_2^2}{2\delta^2(1 + \lfloor \frac{L-P+1}{M} \rfloor)} \approx 2 + \frac{P\|\hat{\mathbf{K}}_{P-1}\|_2^2}{8\pi^2 k_m^2/M^2(1 + \lfloor \frac{L-P+1}{M} \rfloor)} \\ &= 2 + O\left(\frac{M^3}{L}\right). \end{aligned} \tag{3.91}$$

Meanwhile, when L is very large, and thus $\delta(L+1) > 2(P-1)$ is satisfied, Bazán's bound in Equation (3.88) *scales with $1 + O\left(\sqrt{M}/\sqrt{L}\right)$ for $L/M \gg 1$. Thus, to retain the same upper bound of condition number, one only needs to increase the number of time delays at the* same *same rate as the sampling.*

Figure 3.5 shows that the residuals from the least squares problem in both the time and spectral domains decrease exponentially with the addition of time delays. Further, the a posteriori MSE shows significant improvement with the addition of time delays.

Figure 3.6 shows the trend of the 2-norm condition number in both the time and

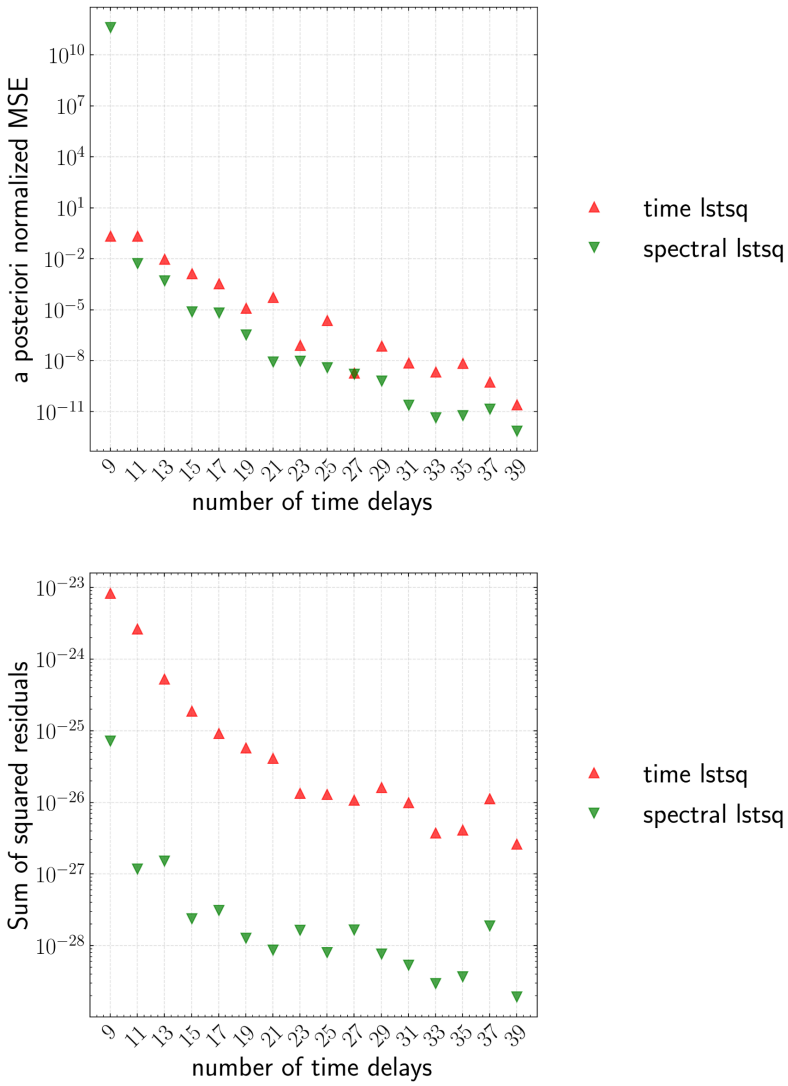


Figure 3.5: Effect of time delay L on $M = 500$ oversampling case. Top: A posteriori MSE normalized by standard deviation of $x(t)$ with increasing time delays. Bottom: Sum of squared residuals with increasing time delays.

spectral domains. The condition number decays exponentially in the spectral case, but increases in the time domain case. This appears to be contradictory since the condition number is typically reflective of the quality of the solution. However, since SVD regularization is implicit in `scipy.linalg.lstsq` with `gelsd` option, computing the 2-norm condition number in the same way as in the numerical solver, i.e., effective condition number [4] is a more relevant measure of the quality of the solution of the SVD truncated system. Thus, the reasons for improved predictive accuracy are due to a) the increasing dimension of the solution space for a potentially under-determined system with more time delays, and b) the well conditioned system after SVD truncation as shown in Figure 3.6. The large condition number in the time domain with increasing number of delays is a result of the ill-conditioning of $\mathbf{V}_P(\omega^{k_0}, \ldots, \omega^{k_{Q-1}})$ and $\mathrm{diag}(a_{i_0}, \ldots, a_{i_{P-1}})$ in Equation (3.37).

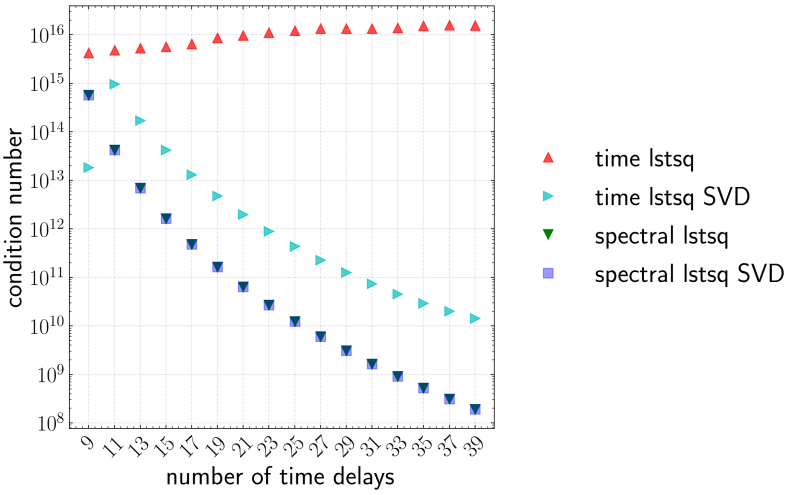


Figure 3.6: $M = 500$ oversampling case: effective condition number decreases with increasing time delay L

[4] i.e., SVD with the same thresholding ($\epsilon = 10^{-15}$) such that any singular value below $\epsilon \cdot \sigma_{max}$ is removed

3.4.7 Effect of subsampling on model performance

As indicated in Remark 3, reducing the number of samples per period M is shown to decrease the upper bound on the condition number. For a given signal, however, there is a restriction on the minimum possible M compared to the number of time delays L. In the above case for the 5-mode sine signal, $i_{\frac{P}{2}-1} = 12$, and thus the minimal sampling per period that one can use to perfectly preserve the original signal in the subsampling is $M = 26$. The condition number with M ranging from 26 to 98 is shown in Figure 3.7. This shows the effectiveness of subsampling in reducing the condition number significantly. Correspondingly, the a posteriori normalized MSE is also reduced as shown in Figure 3.7.

The previous two subsections demonstrated the role of numerical conditioning on model performance. We note that explicit stabilization techniques (*Le Clainche and Vega*, 2017; *Champion et al.*, 2019) require further investigation.

3.4.8 Analysis of noise effect with pseudospectra

Note that our analysis and experiments thus far have been based on noise-free assumptions. When additive noise is present in the data, the *minimal* number of time delays as given by the results in Section 3.3 can be optimistic as we will confirm shortly. Alternatively, one might de-noise the data as by using for instance, optimal SVD thresholding *Gavish and Donoho* (2014) for the delay matrix with i.i.d. Gaussian noise. To illustrate the effect of noise, the toy 5-mode sine signal in Section 3.5.1.1 is considered, but the training horizon is increased to one complete period of data. Consider additive i.i.d. Gaussian noise with signal-to-noise ratio (with respect to the standard deviation) of 1%. To assess the influence of noise rigorously, we take an ensemble of 500 data trajectories and train a linear model with ordinary least squares on such data. In other words, for each sample trajectory, we have a slightly perturbed linear model associated with the data. The influence of noise is evaluated

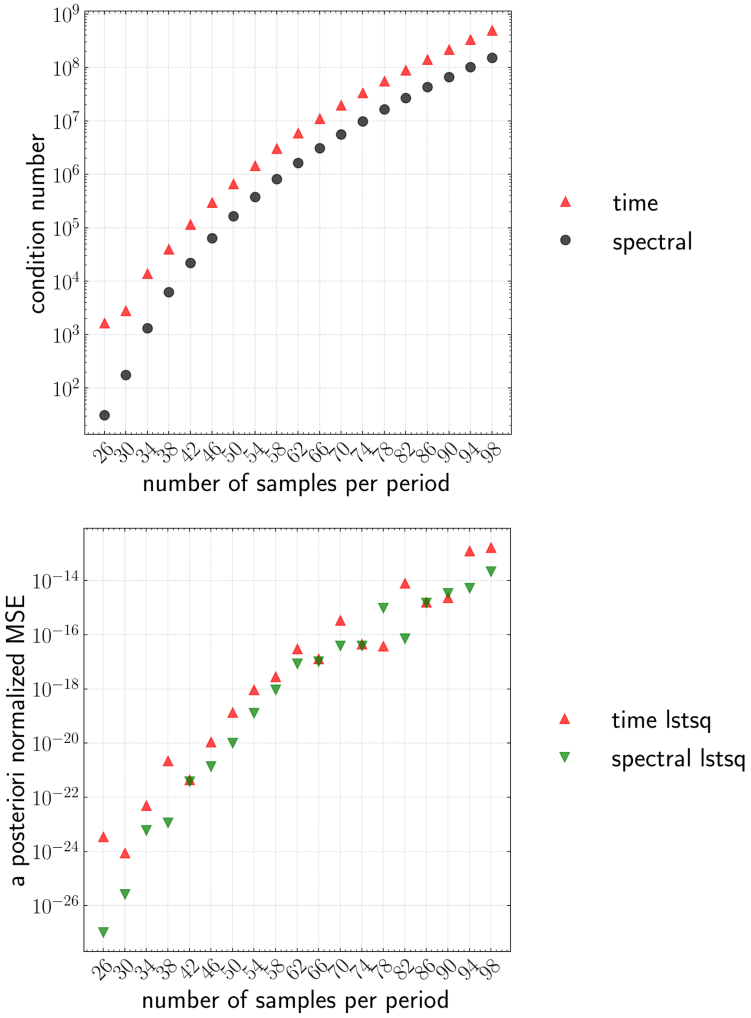


Figure 3.7: Top: Condition number as a function of sampling rate. Bottom: A posteriori normalized MSE with sampling rate.

in the resulting distribution of eigenvalues (a priori sense) and long-time predictions (a posteriori sense). As shown in Figures 3.8 and 3.9, the theoretical optimality of $L = 9$ does not hold as the model becomes overly dissipative. Instead, $L = 20$ is required to have a reasonable prediction. It should be noted that the noise in the training data is too small to be observed in Figure 3.9, while the impact on the linear model is significant, as represented from the red shaded region. Moreover, as L increases, it is observed that the "cloud" of eigenvalues shifts from the left half plane towards the imaginary. Interestingly, the "clouds" associated with spurious modes are much more scattered than those of the exact modes on the imaginary axis, i.e., the spurious modes are *more sensitive* to the noise in the data. As L becomes increasingly large, e.g., $L = 39$, those clouds merge together along the imaginary axis, resulting in higher uncertainty due to the possibility of unstable modes. This is also reflected in the a posteriori predictions in Figure 3.9. Interestingly, the ensemble average of a posteriori prediction appears to show better predictions, even though each individual prediction can be divergent. This implies that an appropriate Bayesian reformulation could make the model more robust to noise *Pan and Duraisamy* (2020c).

Next, we will analyze the robustness of the linear time delayed model with respect to noise in a more general sense. Recall that the previous analysis on condition number in Section 3.4.6 with periodic assumptions indicates robustness to noise with increasing time delays. For a more stringent description of the robustness, we introduce the concept of *pseudospectra Trefethen et al.* (1993). Here we define the ϵ-pseudospectra of the block companion matrix $\mathbf{A}_L$ in Section 3.4.1 as Λ_ϵ in Equation (3.92).

$$\Lambda_\epsilon(\mathbf{A}_L) = \{z \in \mathbb{C} : \sigma_{\min}(z\mathbf{I} - \mathbf{A}_L) \leqslant \epsilon\}, \tag{3.92}$$

where $\sigma_{\min}$ represents the minimal singular value. As shown in Figure 3.10, it is observed that the robustness of the solution decreases the increasing L and becomes most sensitive to noise at the noise-free optimal $L = 9$, following which the robust-

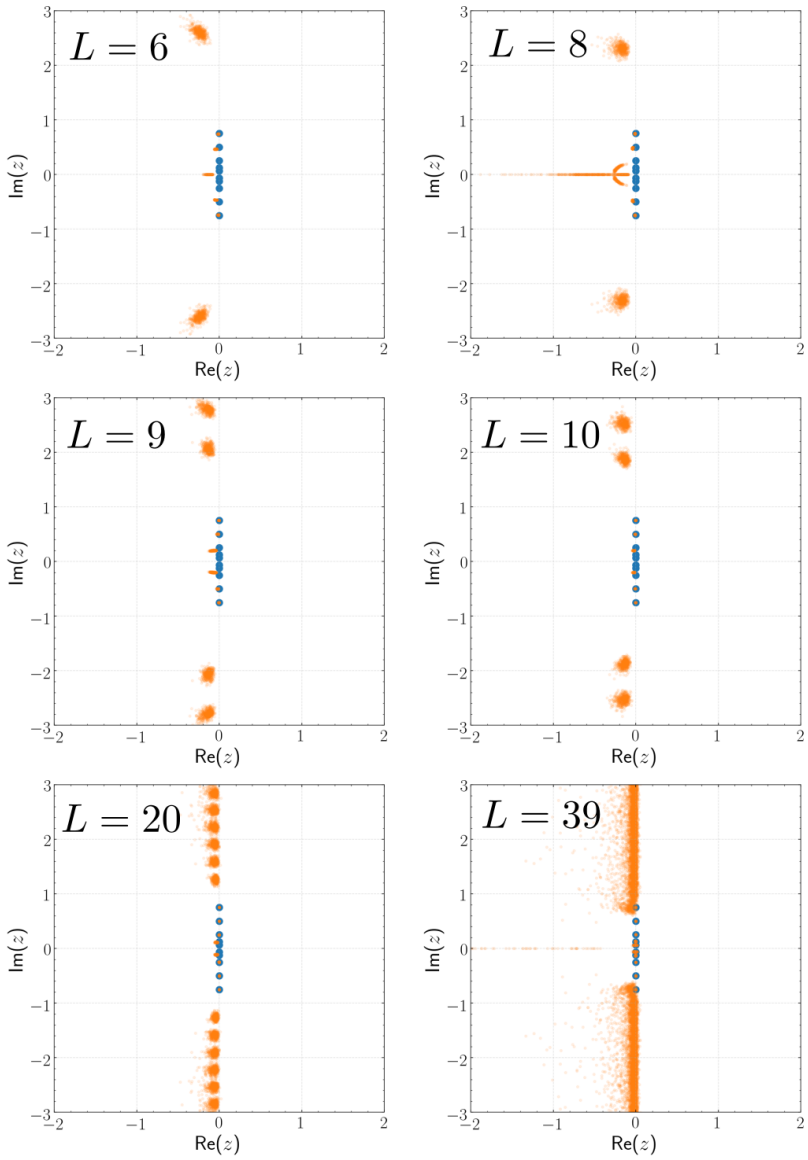


Figure 3.8: Eigenvalue distribution of linear model from noisy data with signal-to-noise ratio as 0.01 (orange) and noise-free data (blue). Time delay ranges from $L = 6$ to $L = 39$.

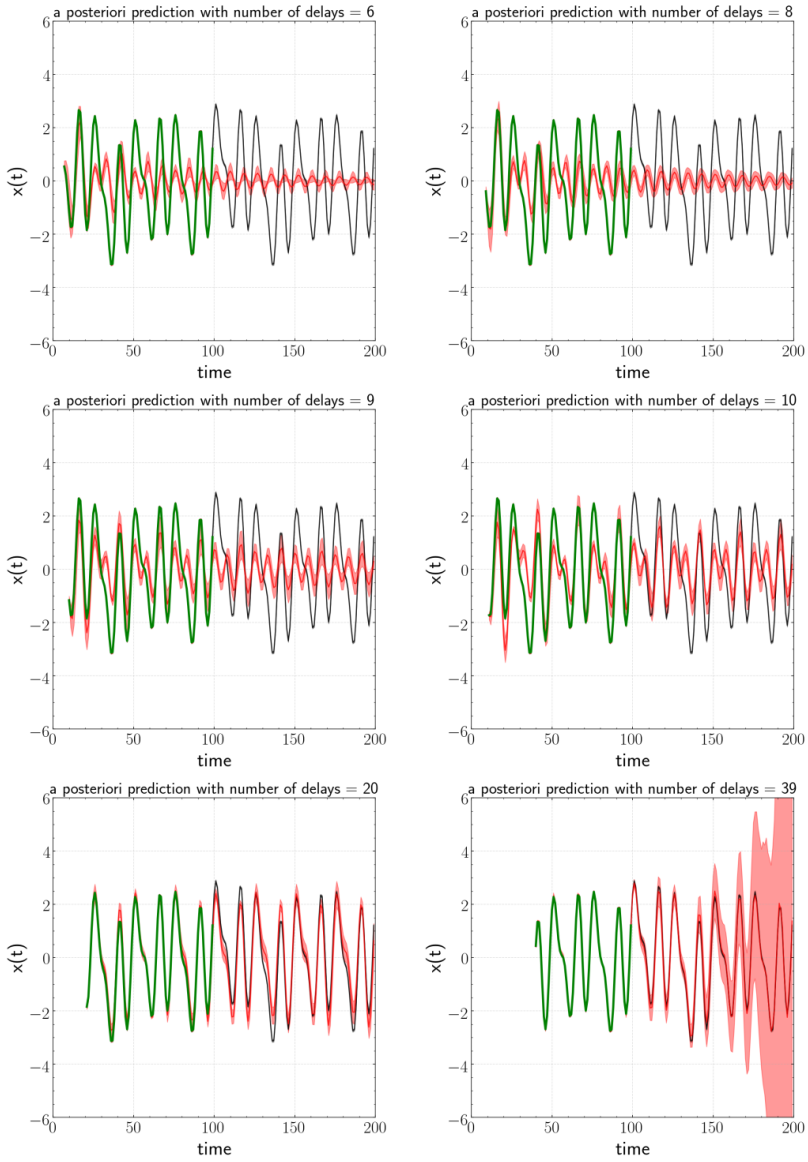


Figure 3.9: A posteriori prediction from noisy data with signal-to-noise ratio of 0.01. Green: training data. Black: whole data. Red: prediction from linear model. Shaded regions represents the uncertainty range of ± 2 standard deviations. Note that all of training, whole and predictions contain shaded region but the noise on training/whole data is too small to be observed.

ness improves as L increases, which is consistent with previous analysis on condition number.

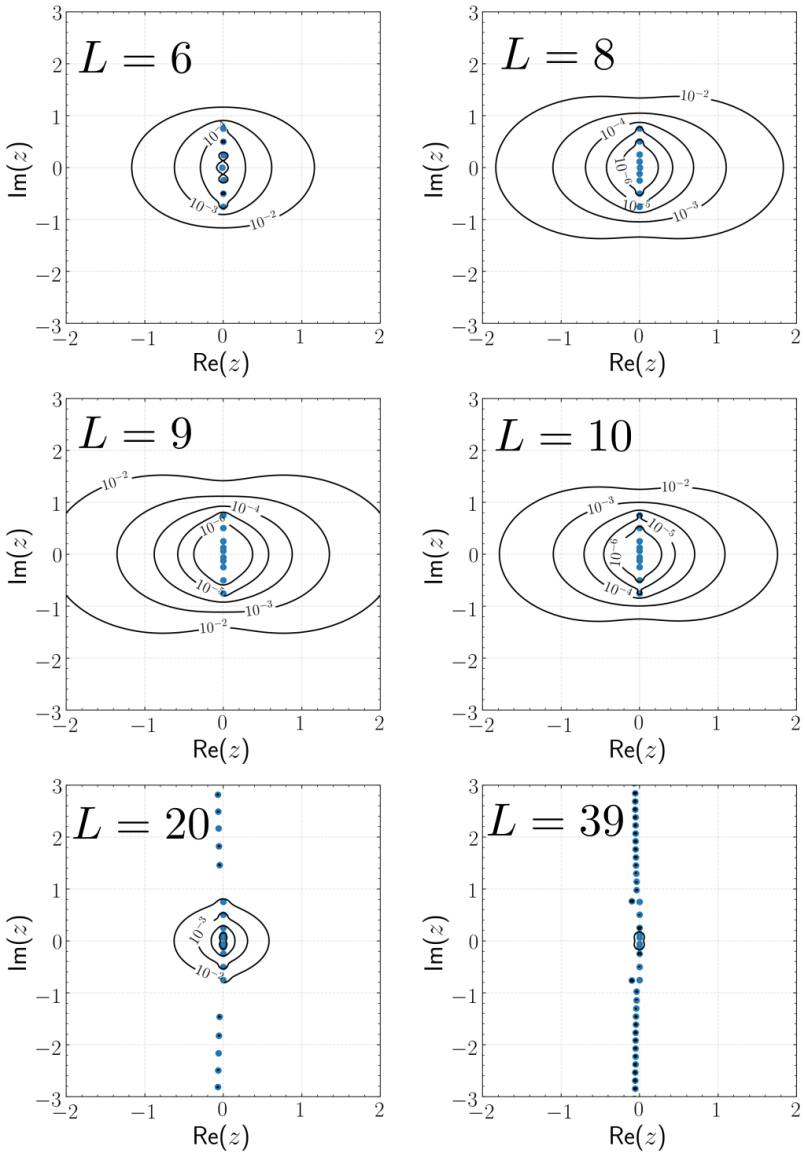


Figure 3.10: Isocontours of pseudospectra at $\epsilon = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ for different time delays L for the toy 5 waves case.

3.4.9 Appropriate time delay for large-scale chaotic dynamical systems

Linear time delayed models have been investigated for chaotic dynamics on an attractor (for instance, *Brunton et al.* (2017)). The main challenges are two fold:

a) Chaotic systems may require an infinite number of waves to resolve the continuous Koopman spectrum (*Mezić*, 2005), and b) Practical chaotic systems of interest in science and engineering science are large-scale. For example, realistic fluid flow simulations, may be very large even after dimension reduction, especially for advection-dominated problems (*Lee and Carlberg*, 2020). This would further limit the expressiveness of linear models with time delay.

To illustrate this, consider dimension reduction using SVD on the trajectory data $\{\mathbf{x}_j\}_{j=0}^{M-1}$. One can extract a reduced r-dimensional trajectory, $\{\hat{\mathbf{x}}_j\}_{j=0}^{M-1}$, i.e.,

$$\begin{bmatrix} \mathbf{x}_0 & \dots & \mathbf{x}_{M-1} \end{bmatrix} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top, \quad \hat{\mathbf{x}}_j = \mathbf{U}_r^\top \mathbf{x}_j \in \mathbb{R}^r. \tag{3.93}$$

Recalling Equations (2.29) and (2.30), we have a similar analytic SVD-DMD solution on the time delay data matrix of the reduced r-dimensional system, i.e.,

$$\widehat{\mathbf{A}}_L = \mathbf{Q}_{r'}^\top \mathbf{U}_r^\top \begin{bmatrix} \mathbf{h}_{L+1} & \dots & \mathbf{h}_{M-1} \end{bmatrix} \mathbf{Z}_{r'} \mathbf{\Sigma}_{r'}^{-1} \in \mathbb{R}^{r' \times r'}, \tag{3.94}$$

with the following $r'-$SVD regularization purely for numerical robustness

$$\mathbf{U}_r^\top \begin{bmatrix} \mathbf{h}_L & \dots & \mathbf{h}_{M-2} \end{bmatrix} \approx \mathbf{Q}_{r'} \mathbf{\Sigma}_{r'} \mathbf{Z}_{r'}^\top. \tag{3.95}$$

Note that $\mathbf{A}_L = \mathbf{Q}_{r'} \widehat{\mathbf{A}}_L \mathbf{Q}_{r'}^\top \in \mathbb{R}^{r(L+1) \times r(L+1)}$ with $\text{rank}(\mathbf{A}_L) = r'$. Following the notations of the mode decomposition in Section 2.4.2, we have

$$\mathbf{x}_{k+1} \approx \sum\nolimits_{i=1}^{r'} \lambda_i^{k+1-L} \mathbf{U}_r \mathbf{q}_i \mathbf{p}_i^\top \mathbf{h}_L, \tag{3.96}$$

where $\mathbf{U}_r \mathbf{q}_i$ and $\{\lambda_i^{k+1-L} \mathbf{p}_i^\top \mathbf{h}_L\}_{k=0}^{M-2}$ are the spatial and temporal modes respectively.

Now we can describe the constraints on the maximal number of modes in the

linear model $r^{'}$ from the time delay L. From the restrictions on matrix rank, we have

$$r \leqslant \min\{n, M\}, \quad r^{'} \leqslant \min\{r(L+1), M-1-L\}, \tag{3.97}$$

as illustrated in Figure 3.11. Clearly, we see the maximal number of waves $r^{'}$ stops increasing after the time delay L surpasses the intersection point where $L_* = \frac{M}{r+1} - 1$, $r^{'}_* = \frac{r}{r+1}M$. This relation indicates that keeping more POD modes in the dimension reduction increases the upper limit of the number of waves in the resulting linear models. The corresponding time delay would decrease with respect to the peak. Interestingly, for $L > \frac{M}{r+1} - 1$, called "overdelay", might yield an underdetermined linear system as in Equation (2.30). For example, we can choose $L_{opt} = \lceil \frac{M}{r+1} \rceil$. The solution of that system would, however, result in a least square residual near machine precision, leading to overfitting even in a posteriori sense. Note that practical problems may require denoising on the trajectory data.

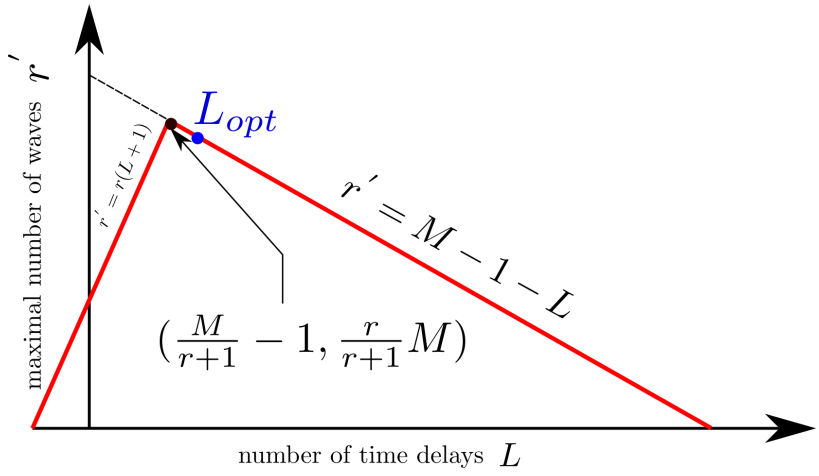


Figure 3.11: Constraints on maximal number of waves $r^{'}$ in the linear model with time delays.

Table 3.1: Summary of the structure of time delay embedding for VdP.

	P	L	$i_{P/2-1}$	M_{min}
$\tilde{x}_1(t)$	10	9	9	20
$\tilde{x}_2(t)$	18	17	18	38
$\tilde{x}_{1,2}(t)$		8		38

3.5 Applications

3.5.1 Van der Pol oscillator

Now we consider the Van der Pol oscillator (VdP) with forward Euler time discretization:

$$\begin{bmatrix} x_1^{n+1} \\ x_2^{n+1} \end{bmatrix} = \begin{bmatrix} x_1^n \\ x_2^n \end{bmatrix} + \Delta t \begin{bmatrix} x_2^n \\ \mu(1 - x_1^n x_1^n)x_2^n - x_1^n \end{bmatrix}, \tag{3.98}$$

where $\mu = 2$, $x_1^0 = 1$, $x_2^0 = 0$, $\Delta t = 0.01$. After 530 time steps, the system approximately falls on the attractor with an approximate period of 776 steps. Total data is collected after the system falls on the attractor for 4 periods.

As shown in Figure 3.12, Fourier spectrum for each component of VdP system shows that the exhibition of an approximate sparse spectrum with $P = 10$ and $P = 18$ for x_1 and x_2 respectively. As indicated from Theorem III.5, the corresponding time delay and minimal sampling rate is summarized in Table 3.1.

3.5.1.1 Extrapolation: scalar case

From Table 3.1, it is clear that the smallest number of samples per period is significantly smaller than the original number of samples per period, i.e., $M = 776$. The analysis in the previous section also showed that the choice of a smaller number of samples per period is helpful in reducing the condition number. Thus, we choose a moderately subsampled representation without any loss in reconstruction compared to the filtered representation. Individually treating the first and second components,

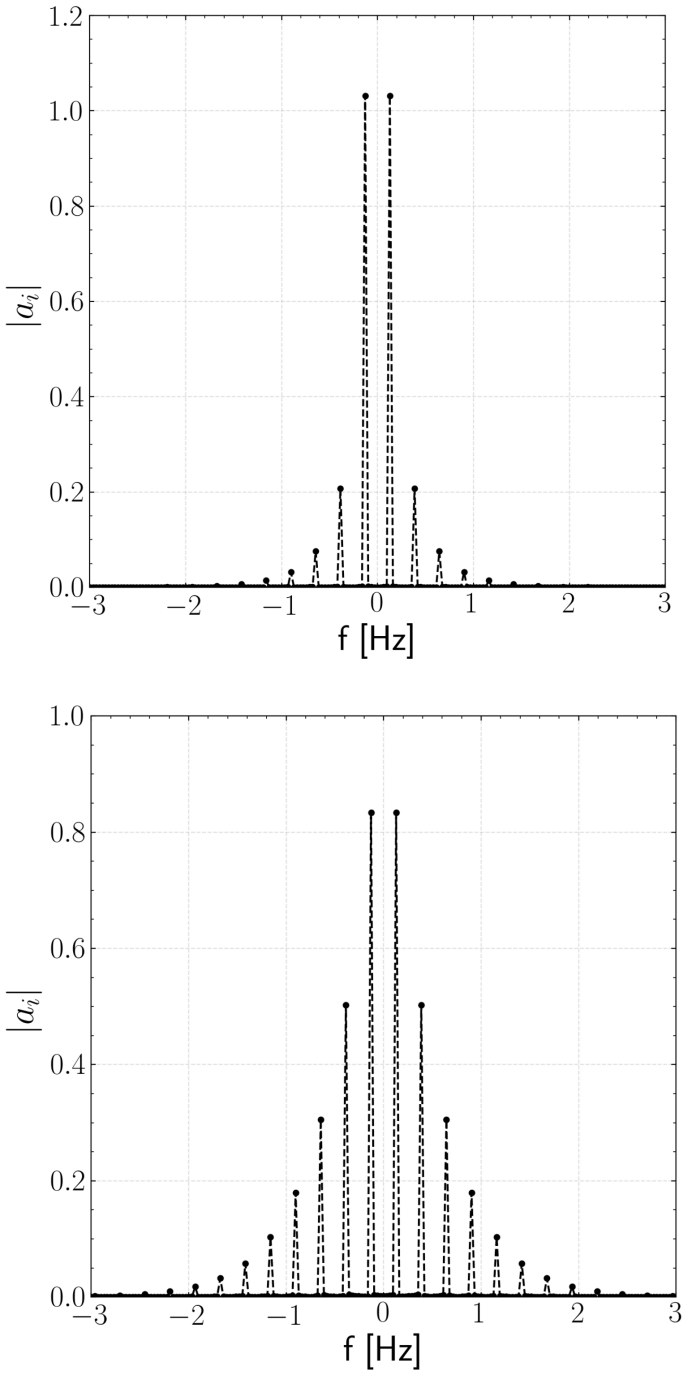


Figure 3.12: Fourier spectrum for VdP system. Top: x_1. Bottom x_2.

we choose $M = 200, 100$ with theoretical minimum time delays $L = 9, 17$, respectively.

Numerical results displayed in Figure 3.13 show that, even using training data that covers less than 25% of the period for the first component, and 50% of the period for the first component, the linear model with minimal time delays is still able to accurately predict the dynamics over the entire time period of the limit cycle. Note that a similar predictive performance is expected for the original (unfiltered) VdP system.

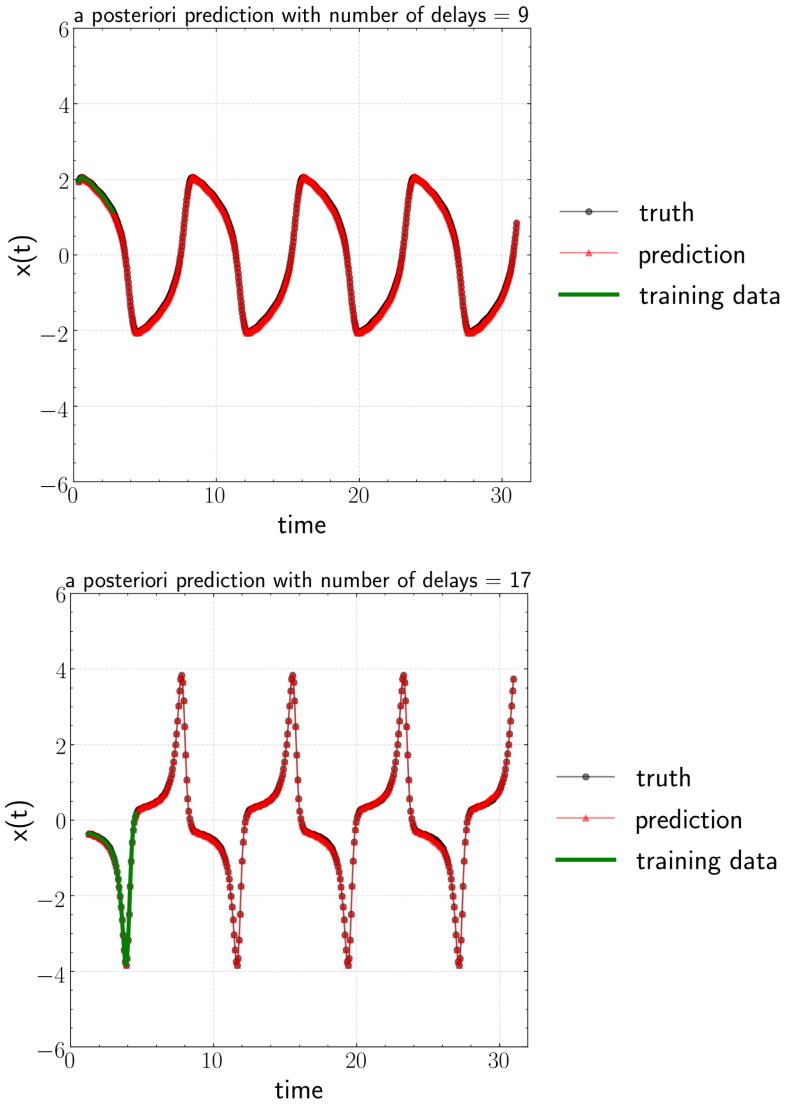


Figure 3.13: Prediction vs ground truth for each component of VdP. Top: first component. Bottom: second component.

3.5.1.2 Extrapolation: vector case

As given in Table 3.1, Lemma III.6 predicts that the consideration of both components requires only 8 delays. The effectiveness of the criterion developed in Lemma III.6 is confirmed to a resounding degree in Figure 3.14. The top figure shows the predictive performance of the time delayed linear model for the minimum number of delays and the bottom figure shows the behavior of the a posteriori normalized MSE versus the number of time delays. It should be recognized that in contrast to the scalar case, in which the minimal time delay can be directly inferred from the Fourier spectrum, the vector case requires *iterative* evaluations of the rank test in Lemma III.6.

3.5.2 Quasi-periodic signal

As indicated in Laudau's route to chaos *Landau* (1944), quasi-periodic systems play an important role in the transition from a limit cycle to fully chaotic flow.We consider the following quasi-periodic signal

$$x(t) = \cos(\sqrt{2}t/2)\sin(\sqrt{3}t/2)\cos(t), \tag{3.99}$$

where $t \in [0, 40]$. Consider a sampling interval $\Delta t = 0.1$, we consider the linear model trained on the first 60 snapshots, i.e., $t \in [0, 6]$.

As shown in Figure 3.15, the linear model with $L = 7$ accurately predicts the future state behavior of the quasi-periodic system with only a fraction of data limited in the range $[-0.25, 0.55]$ while the whole data ranges from $[-0.944, 0.902]$. Indeed, the minimal time delay $L = 7$ is determined by the number of frequencies in the signal. The analysis on the minimal number of time delays for scalar time series as in Section 3.3 can be extended to quasi-periodic system. Consider the trigonometric

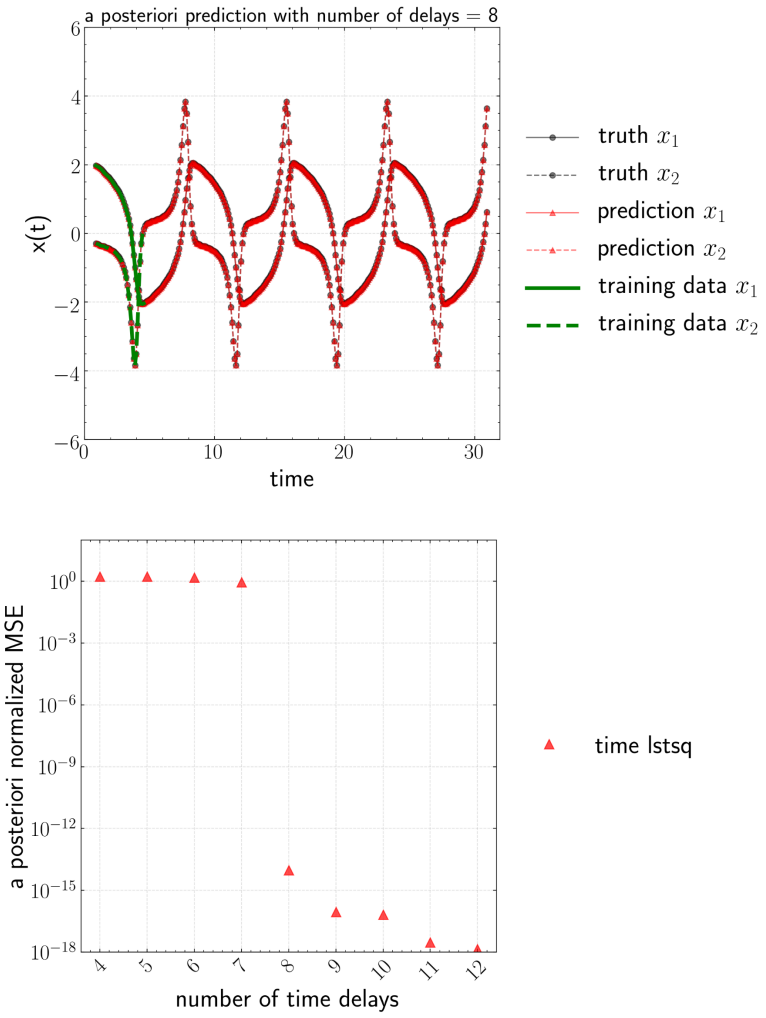


Figure 3.14: Top: Prediction vs ground truth with $M = 80$ for VdP system. Bottom: A posteriori MSE normalized by standard deviation with as a function of the number of time delays for the vector case.

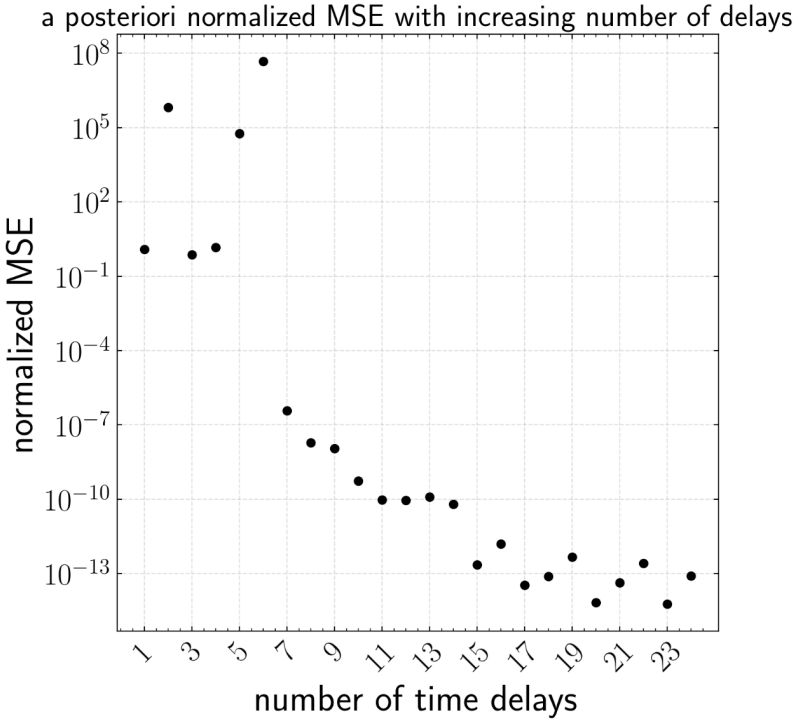


Figure 3.15: Top: Prediction vs ground truth for the toy quasi-periodic signal. Bottom: A posteriori MSE normalized by standard deviation with as a function of the number of time delays.

identity, we have the following equivalent equation of Equation (3.99),

$$x(t) = \frac{1}{4}\Big(\sin\Big(\frac{(\sqrt{2}+\sqrt{3}+2)t}{2}\Big) + \sin\Big(\frac{(\sqrt{2}+\sqrt{3}-2)t}{2}\Big) - \sin\Big(\frac{(\sqrt{2}-\sqrt{3}+2)t}{2}\Big) - \sin\Big(\frac{(\sqrt{2}-\sqrt{3}-2)t}{2}\Big)\Big).$$

Therefore, we require $L = P - 1 = 7$ time delays to fully recover the signal which is confirmed in Figure 3.15.

3.5.3 3D Turbulent Rayleigh-Bénard convection

As a final test case, we consider Rayleigh-Bénard convection, which is a problem of great interest to the fluid dynamics community. As displayed in Figure 3.16, the fluid is confined between two infinite horizontal planes with a hotter lower plane. The Rayleigh number, which represents the strength of buoyancy with respect to momentum and heat diffusion is defined as $Ra = U_f^2 H^2/\nu\kappa = \alpha g \Delta T H^3/\nu\kappa$ where α is the thermal expansion coefficient, κ is the thermal diffusivity, ΔT is the temperature difference between hot and cold planes, and $U_f \triangleq \sqrt{\alpha g \Delta T H}$ is the so-called free-fall velocity of a fluid parcel. Additional parameters that govern the dynamics are aspect ratio $\Gamma \triangleq L/H$, the Prandtl number $Pr = \nu/\kappa$. L is the horizontal length scale of the domain. The computational domain is taken as a rectangular box with periodic side walls. We set $Ra = 10^7$ for fully turbulence; $H = \pi L_x = \pi L_y$ and $Pr = 1$. This domain is discretized uniformly in x and y direction with 128×128 grid points and in z direction with 128 grid points highly refined near the wall. The thickness of thermal boundary layer is sufficiently resolved *Verzicco and Camussi* (2003) since $\delta_\theta/H \sim 1/2Nu \approx 10\Delta z$, where Δz is the grid size in z direction closest to the wall.

The simulation is performed by solving 3D incompressible Navier-Stokes equations with a Boussinesq approximation using OpenFOAM *Jasak et al.* (2007). Linear heat conduction, i.e., an unstable equilibrium state is set as initial condition. The

simulation is performed over four thousand characteristic advection time units, approximately $1.264\tau_{\text{diff}}$, where $\tau_{\text{diff}} \triangleq H^2/\nu$, $\tau_{\text{adv}} \triangleq \sqrt{H/\alpha g \Delta T}$. The sampling interval is $\Delta t = 4\tau_{\text{adv}}$. Note that this dynamical system contains approximately 2 million degrees of freedom. Here we perform dimension reduction on the sampled system state u, v, w, T similar to *Pan et al.* (2020). First, normalization for each component and mean subtraction is performed. Second, as shown in the bottom subfigure in the Figure 3.16, more than 99% of variance for the non-linear system is retained in the first $r = 800$ POD modes on the normalized data. After removing the effect of initial condition (the first 100 snapshots), we use 900 snapshots *Pan and Arnold-Medabalimi* (2020) for analysis.

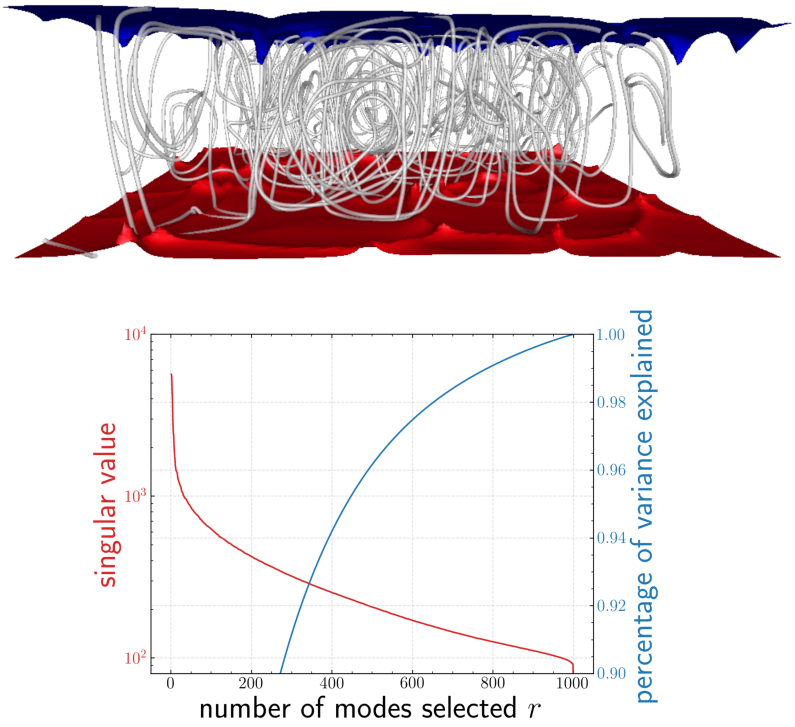


Figure 3.16: Top: Iso-surfaces of temperature at $T = 295$ (red) and $T = 285$ (blue) with streamlines of velocity field (grey) at $t = 7.28$ for the Rayleigh-Bénard turbulent convection at $Ra = 10^7$. Bottom: Singular value distribution and percentage of variance explained.

We consider the first 800 out of 900 snapshots as training data. Then we perform

a posteriori evaluation for 900 steps to examine the reconstruction performance and predictions on future time steps. As shown in Figure 3.17, performing SVD-DMD ($L = 0$) on this dataset with $r = 800$ results in a set of unstable eigenvalues, leading to undesired blow up in a posteriori evaluation after $180\Delta t$. While the model with time delay $L = 1$, overfits to the training data from 0 to approximately $800\Delta t$, it yields stable predictions. Note that in this case $L_{opt} = \lceil \frac{M}{r+1} \rceil = 1$.

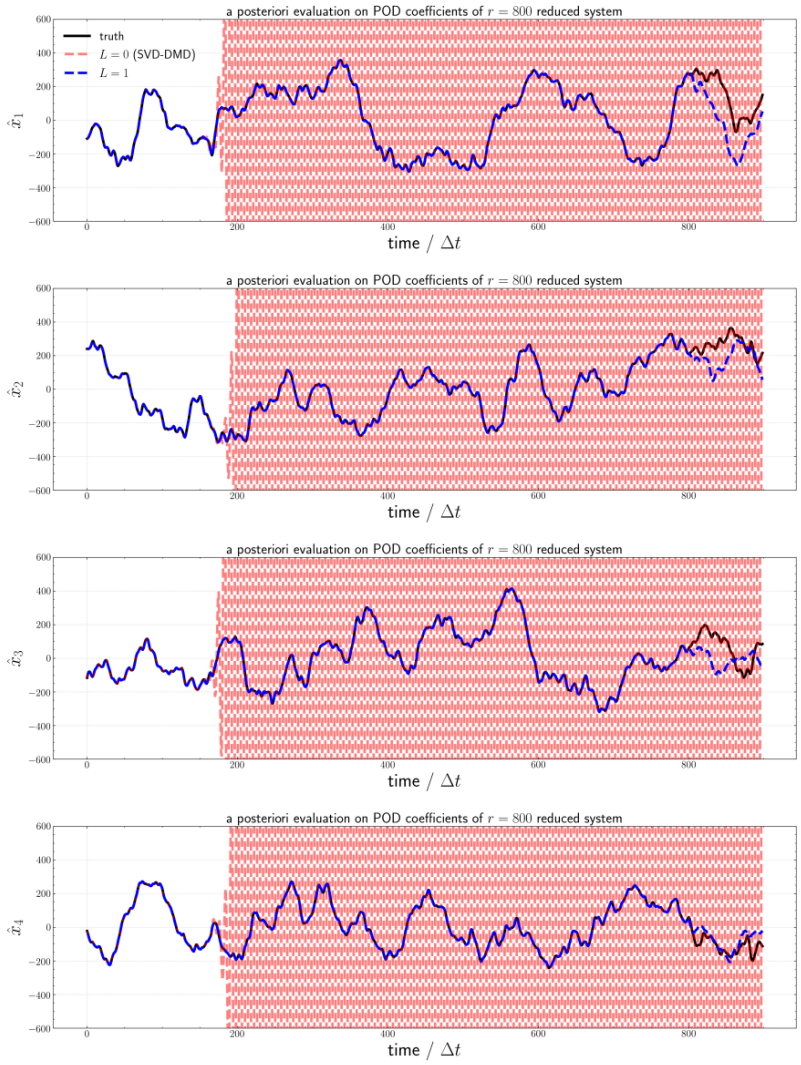


Figure 3.17: Comparison of a posteriori evaluation between linear model without/with time delay $L = 1$ for the reduced system with $r = 800$. Note that $0 \leqslant t \leqslant 800$ is training horizon while $800 < t \leqslant 900$ is testing horizon.

We then take the entire 900 snapshots trajectory as training data to investigate the impact of of time delays L on stabilizing the reconstruction at various r. As shown in Figure 3.18, we first observe that as r decreases, the numerical condition number increases simply as a consequence of retaining more small singular values. Secondly, we observe a general trend that, for each r, model performance worsens as L increases from 0 to $L_{opt}-1$, i.e., the transient point where linear systems approximately change from over-determined to under-determined. For the current data specifically, we observe that the system becomes stable as L increases as the system becomes under-determined. Thirdly, we observe that the condition number shares a similar pattern with the reconstruction performance for each r.

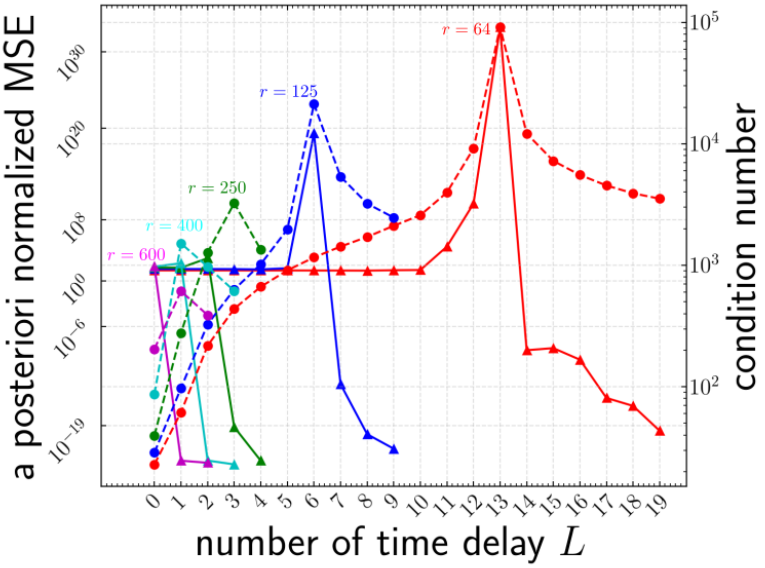


Figure 3.18: Dependency of model reconstruction performance and condition number on the number of time delays L with varying reduced dimension r for turbulent Rayleigh-Bénard convection. Solid line: normalized mean-squared-error. Dashed line: condition number.

3.5.4 2D Single-Injector Combustion Process with Input

Constructing efficient models for optimization, design, uncertainty quantification, and control of combustion processes is an active research area. The compressible,

chemically reacting and turbulent flows present formidable challenges in intrusive ROMs (*Huang et al.*, 2018, 2020b). In this section, we alternatively consider the use of time delay DMD with input (TD-DMDc) to construct ROMs of the combustion problem shown in fig. 3.19.

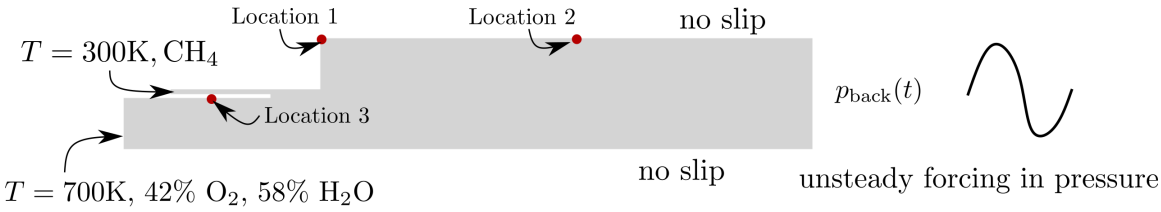


Figure 3.19: Setup of 2D combustion process

3.5.4.1 Full Order Model

The Full order simulation (FOM) is performed by solving the 2D compressible N-S equations using the GEMS solver (*Harvazinski et al.*, 2015). The following one-step combustion reaction with finite-rate chemistry is employed:

$$\text{CH}_4 + 2\text{O}_2 \rightarrow \text{CO}_2 + \text{H}_2\text{O}. \tag{3.100}$$

Non-reflective boundary conditions are imposed on the exit while a constant mass flow is imposed at the inlet. An unsteady forcing in pressure shown in eq. (3.101) is imposed at the exit.

$$p_{\text{back}}(t) = p_{\text{back,ref}}(1 + A\sin(2\pi f t)), \tag{3.101}$$

where $p_{\text{back,ref}} = 10^6$ Pa, $A = 0.1$, $f = 5000$ Hz. The FOM is simulated from 15ms to 17ms. More details can be found in *Swischuk et al.* (2020). The above test problem has been used as a benchmark (*Swischuk et al.*, 2020) using a regularized multivariate quadratic polynomial regression approach on the POD coefficients. This technique is termed *operator-inference*.

3.5.4.2 Modeling

After uniformly sampling with the SVD-reduced data collected as $\{\mathbf{x}_j\}_{j=0}^{M-1}$, at time t_j, we have the following expression for TD-DMDc in eq. (3.102) (which can be viewed as a natural extension to DMDc (*Proctor et al.*, 2016)):

$$\hat{\mathbf{x}}_{j+1} = \mathbf{W}_0\mathbf{x}_j + \mathbf{W}_1\mathbf{x}_{j-1} + \ldots + \mathbf{W}_L\mathbf{x}_{j-L} + \mathbf{B}\mathbf{u}_j, \tag{3.102}$$

where $\mathbf{u}_j \triangleq \mathbf{u}(t_j) \in \mathbb{R}^{n_i \times 1}$ is the known external forcing at time t_j. Similar to eq. (2.29), it is natural to formulate least-squares regression to determine the parameters:

$$\mathbf{W}_0, \ldots, \mathbf{W}_L, \mathbf{B} = \arg \min_{\mathcal{B}\in\mathbb{R}^{n\times n_i}, \{\mathcal{W}_i\}_{i=0}^{L}\in\mathbb{R}^{n\times n}} \left\| \begin{bmatrix} \mathcal{W}_L & \ldots & \mathcal{W}_0 & \mathcal{B} \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 & \ldots & \mathbf{x}_{M-2-L} \\ \vdots & \vdots & \vdots \\ \mathbf{x}_L & \ldots & \mathbf{x}_{M-2} \\ \mathbf{u}_L & \ldots & \mathbf{u}_{M-2} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_{L+1} & \ldots & \mathbf{x}_{M-1} \end{bmatrix} \right\|_F, \tag{3.103}$$

As in *Swischuk et al.* (2020), our task is to predict the future state from $t \in [16, 17]$ ms given only the data from FOM for $t \in [15, 16]$ ms.

First of all, we construct a set of POD basis from the 10,000 snapshots between 15ms to 16ms with 22 POD modes, which retains 98.5% of the variance. Then, we uniformly downsample the 22-D 10,000 snapshots to 500 snapshots, following the setup from *Swischuk et al.* (2020). First, we find the optimal delay embedding by using the first 400 snapshots as training data and the remaining 100 snapshots as validation data. Note that improved accuracy was noted with a larger time delay in the previous 3D turbulent convection case, because of improved numerical conditioning as stated in section 3.4.6. However, we observed the existence of unstable modes in this problem

even with large time delays. This is perhaps unsurprising given our previous example on a noisy system with all eigenvalues on the imaginary axis in section 3.4.8. Hence, we consider an explicit stabilization procedure, by projecting unstable eigenvalues onto the unit circle. Given the results in fig. 3.20, we select $L = 47$ as it achieves minimal a posteriori MSE.

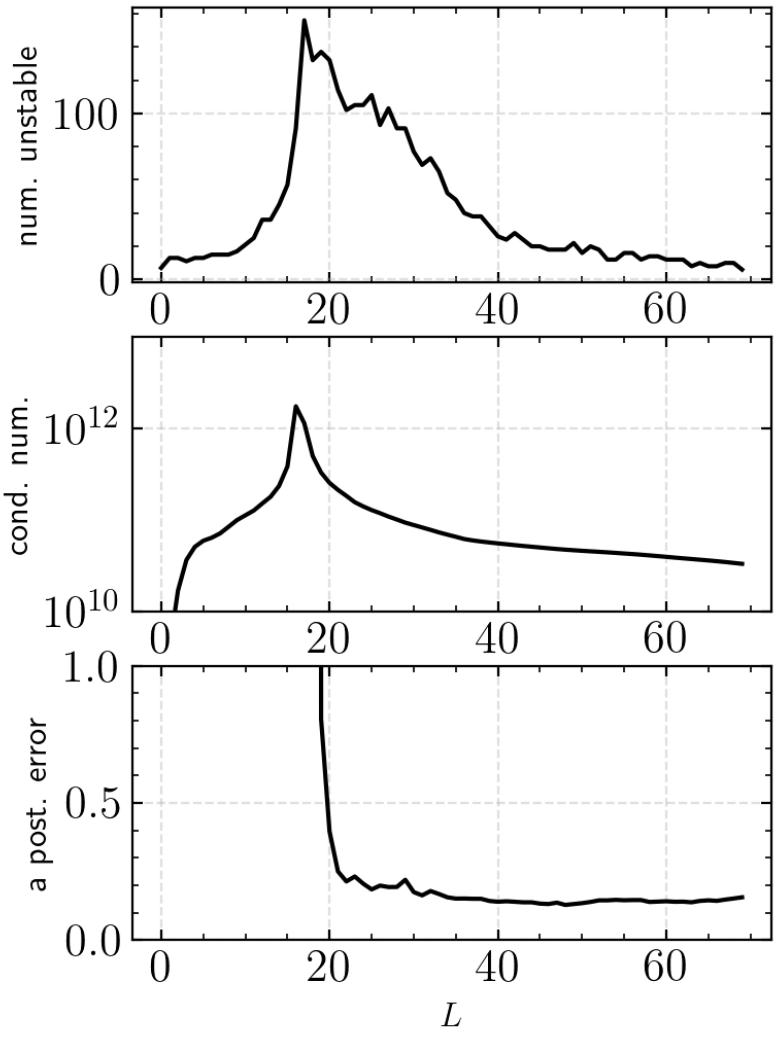


Figure 3.20: Impact of number of time delays L: number of unstable modes (first row); condition number (second row); a posteriori error (third row) on the entire range of data (only initial condition is given).

3.5.4.3 Comparisons against vanilla LSTM-RNN and OpInf

Next, we compare our model against two popular state-of-the-art methods:

- *Operator-inference (OpInf)* (*Peherstorfer and Willcox*, 2016): second order regularized multivariate polynomial regression on POD coefficients. The results are chosen from their best model with $\lambda = 65476$ (*Swischuk et al.*, 2020; *McQuarrie et al.*, 2020), which is computed from their open-source package (*McQuarrie et al.*, 2020).

- *Vanilla LSTM-RNN* (*Hochreiter and Schmidhuber*, 1997). We implement LSTM-RNN using Keras (*Chollet et al.*, 2015) with two sequences returning an LSTM layer with 50-dimensional output units, one LSTM layer with single 100-dimensional output, followed by a linear layer to reconstruct the state. Here we neglect the known input and consider it as a purely data-driven time series prediction task. We normalize the data by subtracting the mean and dividing by the component-wise maximal standard deviation from training data (15ms-16ms). The look-back length (similar to the number of time delay) is chosen as $L = 30$ as it can reconstruct the training data extremely well [5]. We did not observe significant differences in the training data as we further increase the look-back length. We used the Adam optimizer with a learning rate $1e-3$ to train the network.

As displayed in figs. 3.21 to 3.23, we compare predictions of pressure, temperature and mass fraction of methane from three models at three "red dot" monitored locations defined in fig. 3.19. From figs. 3.21 to 3.23, we find that T and $Y_{\mathrm{CH_4}}$ is more difficult to predict than p. Further, we summarize the MSE normalized by the standard deviation of the ground truth for each model in table 3.2. Our TD-DMDc model with over-delay appears to perform slightly better than the two other methods

[5] One of the critical issues with any vanilla neural network model is that they are difficult to reproduce due to the randomness of the stochastic gradient descent.

in predicting the future state of the pressure from $t = 16$ ms to $t = 17$ ms except for the temperature monitored at location 1. Finally, flow-field predictions of pressure, velocity, temperature and mass fraction of methane at a future time 16.5 ms is summarized in fig. 3.24. Again, we find that pressure field is easier to predict than other field, presumably due to the dominance of larger scale coherent structures.

Table 3.2: Normalized MSE from 16ms to 17ms for three models.

Location/p, T	OpInf	LSTM-RNN	TD-DMDc
1, p	0.4318	0.5404	**0.3836**
2, p	0.2127	0.4105	**0.1897**
3, p	0.4619	0.5879	**0.4491**
1, T	**0.0873**	0.1786	0.1807
2, T	3.1453	2.6600	**2.3236**
3, T	0.6333	0.1626	**0.1269**

In terms of model complexity, the number of parameters in OpInf grows *exponentially* with the polynomial order (which is fixed to be second order). The number of model parameters only grows *linearly* with the number of time delays in TD-DMD, with the obvious caveat that the technique requires L steps to be initialized. Note that although OpInf and TD-DMD share the computational cost of least square minimization, selecting the hyperparameter in OpInf requires tuning the L_2 regularization parameter. In TD-DMD, theoretical results indicate that over-delay is encouraged, but again, the precise number has to be tuned. A key benefit of the TD-DMD is that a stable ROM can be guaranteed by removing unstable modes. This is not the case in OpInf or LSTM-RNN. As reported in *Swischuk et al.* (2020), L_2 regularization does not entirely resolve the issue of ROM stability. In addition to the difficulties in choosing appropriate network topologies and higher computational cost, the temporal stability of LSTM and other neural networks is a challenge in learning the evolution of a non-linear dynamical system unless certain structure is imposed as we will see in chapter V.

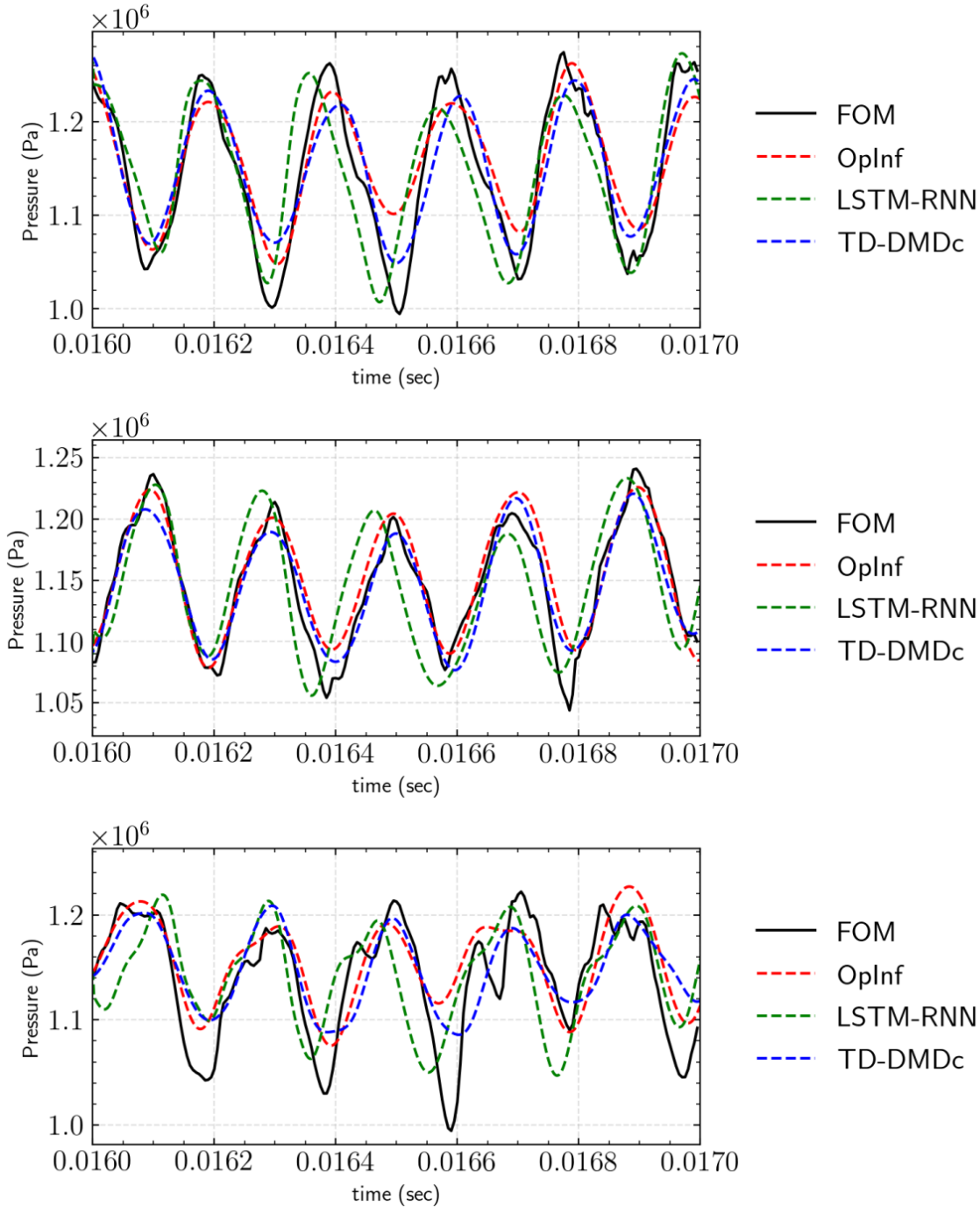


Figure 3.21: Pressure time traces monitored on location 1-3: comparison against *OpInf* model and vanilla LSTM for future prediction from $t = 16$ ms to $t = 17$ ms.

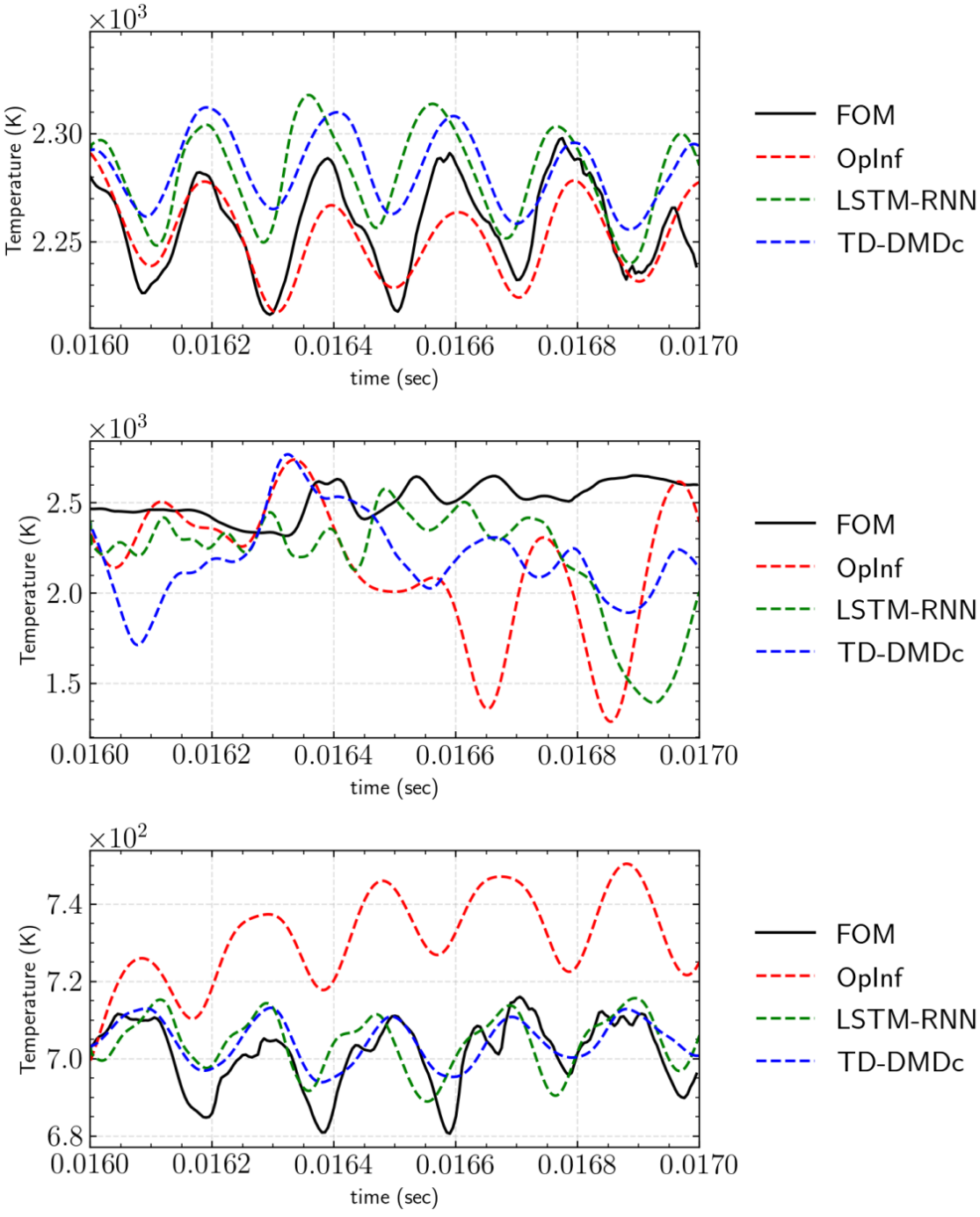


Figure 3.22: Temperature time traces monitored on location 1-3 with comparison against *OpInf* model and vanilla LSTM for future prediction from $t = 16$ ms to $t = 17$ ms.

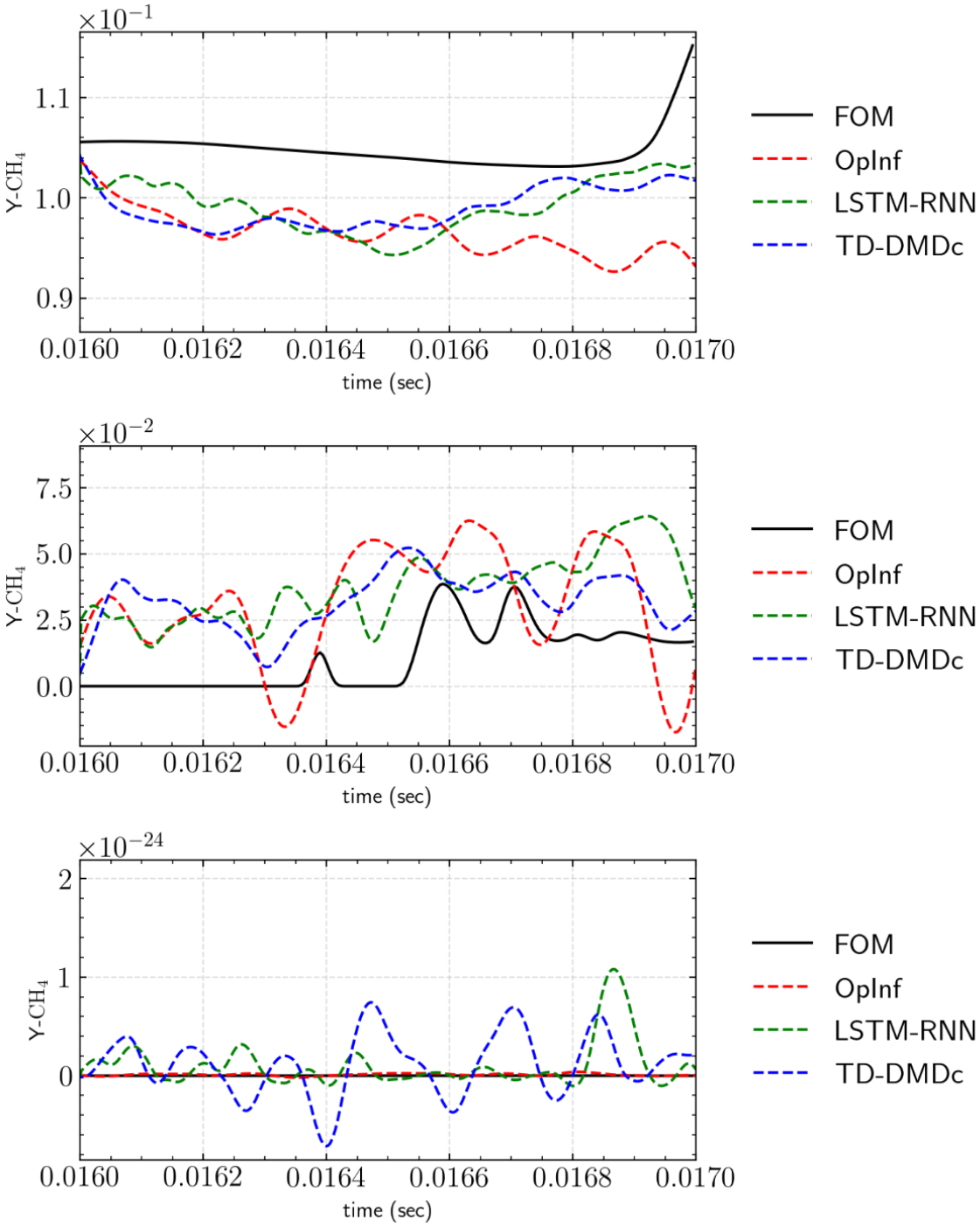


Figure 3.23: Mass fraction of CH_4: time traces monitored on location 1-3 with comparison against *OpInf* model and vanilla LSTM for future prediction from $t = 16$ ms to $t = 17$ ms.

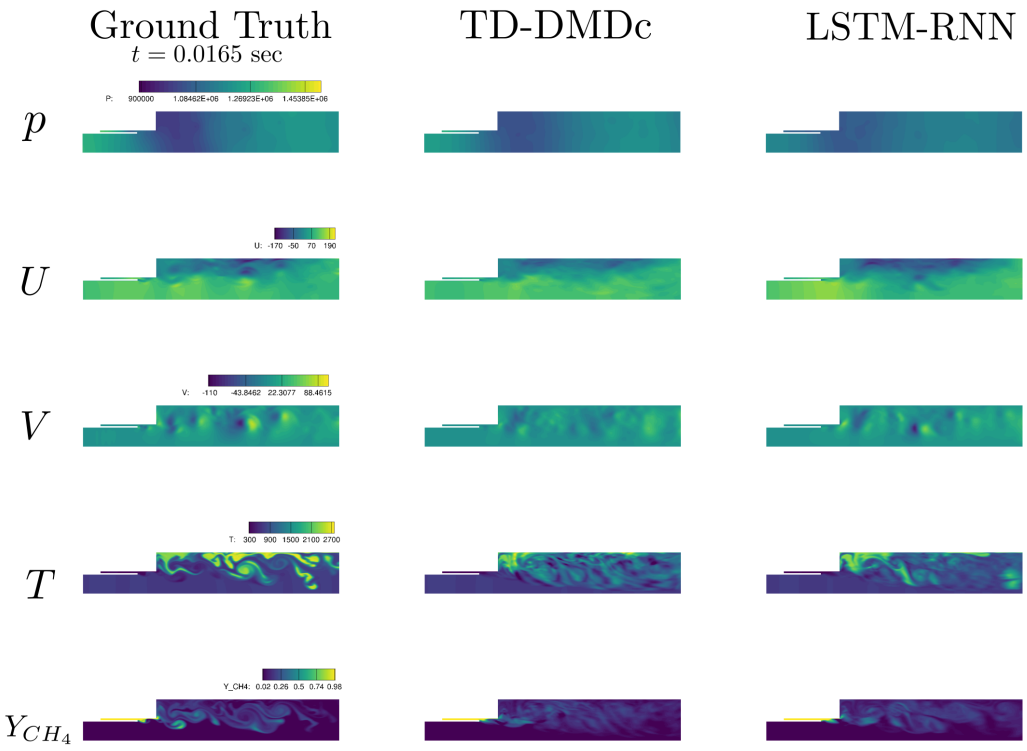


Figure 3.24: Field prediction at $t = 0.0165$ sec from TD-DMDc, LSTM-RNN versus ground truth.

3.6 Summary

This chapter addressed fundamental questions regarding the structure and conditioning of linear time-delay models of non-linear dynamics on an attractor.

The following are the main contributions of this chapter:

1. We proved that for non-linear scalar dynamical systems, the number of time delays required by linear models to recover limit cycles correctly is determined by the sparsity in the Fourier spectrum.

2. In the vector case, we proved that the minimal number of time delays has a tight upper bound that is precisely the output controllability index of a related linear system.

3. We developed an equivalent representation of the linear time-delayed model in the spectral domain and provided the exact solution of the delay transition matrix $\mathbf{K}$ for the scalar case.

4. We derived an upper bound on the 2-norm condition number as a function of the sampling rate and the number of time delays. Thus, ill-conditioning can be mitigated by increasing the number of time delays and/or subsampling the original signal.

5. We explicitly showed that the dynamics over the full period could be perfectly recovered by training the linear time-delayed model over just a partial period, and provided an explanation.

6. Influence of noise is evaluated with ensemble realizations. We further analyzed the stability of the model with the concept of pseudospectra. The results are consistent with our finding on the stabilizing role of the number of time delays.

7. Numerical experiments on simple problems were shown to confirm each of the above theoretical results.

8. The impact of time delays on linear modeling of large-scale chaotic systems (3D turbulent convection and 2D combustion process) was investigated and TD-DMD was confirmed to produce stable and accurate results given enough time delays.

A few observations are pertinent to the above conclusions:

- Due to accuracy considerations on the numerical integrator, the sampling rate in the raw data may be excessively high. We believe that instabilities in prediction arise from choices that lead to poor numerical conditioning. Thus, as an alternative to pursuing explicit stabilization techniques (*Le Clainche and Vega*, 2017; *Champion et al.*, 2019), appropriate sub-sampling and time delays can be employed. Indeed, when a significant amount of noise is present in the data, explicit stabilization, Bayesian inference, or denoising techniques (*Rudy et al.*, 2019) should be warranted.

- The effectiveness of linear time delayed models of non-linear dynamics is that - by leveraging Fourier interpolation - an arbitrarily close trajectory from a high dimensional linear system can be derived. This also intuitively explains the ability of the model - when the signal has a sparse spectrum - to perform "true" predictions without training on a full period of data.

- A key benefit of the TD-DMD is that a stable ROM can be guaranteed by removing unstable modes. This is not the case in OpInf or LSTM-RNN.

CHAPTER IV

Sparsity-Promoting Algorithms to Extract Compact Koopman Invariant Subspaces

> The purpose of computing is insight, not numbers.
>
> *Richard Hamming (1962)*

4.1 Background and Motivation

In general, physical systems governed by PDEs, e.g., fluid dynamics, are infinite-dimensional. From a numerical viewpoint, the number of degrees of freedom can be related to the spatial discretization (for example, the number of grid points). Although a finite-dimensional manifold can be extracted (*Holmes et al.*, 2012), e.g., $O(10)$-$O(10^3)$ dimensions via Proper Orthogonal Decomposition (POD), finding a Koopman-invariant subspace on such manifolds is still challenging. The most popular method to approximate the Koopman operator is the dynamic mode decomposition (DMD) (*Schmid*, 2010; *Rowley and Dawson*, 2017) mainly for two reasons. First, it is straightforward and computationally efficient compared to non-linear counterparts such as Extended DMD (EDMD) (*Williams et al.*, 2015) and Kernel DMD (KDMD) (*Williams et al.*, 2014). Second, the essence of DMD is to decompose a

spatio-temporal field into several temporally growing/decaying traveling/stationary harmonic waves, which are prevalent in fluid mechanics. However, the accuracy of DMD is limited by the assumption that the Koopman-invariant subspace lies in the space spanned by snapshots of the state $\mathbf{x}$. Thus, DMD is used to identify and visualize coherent structures mainly. Indeed, DMD can be interpreted as a L^2 projection of the action of the Koopman operator on the *linear space* spanned by snapshots of the system state (*Korda and Mezić*, 2018a).

To overcome the above limitations, one might naturally augment the observable space with either the history of the state (*Arbabi and Mezic*, 2017; *Brunton et al.*, 2017; *Kamb et al.*, 2018; *Le Clainche and Vega*, 2017) or non-linear observables of the state (*Williams et al.*, 2014, 2015). Time-delay embedding can be very useful in reduced-order modeling of systems for which sparse measurements can be easily obtained, assuming the inputs and outputs are not high dimensional (*Korda and Mezić*, 2018b). Although time-delay embedding is simple to implement and has strong connections to Takens' embedding (*Kamb et al.*, 2018; *Pan and Duraisamy*, 2020b), the main practical issue arises in reduced-order modeling of high fidelity simulations in a predictive setting due to the requirement of a large number of snapshots of the full order model. Further, if one is only interested in the post-transient dynamics of the system state *on* an attractor, linear observables with time delays are sufficient to extract an informative Koopman-invariant subspace (*Arbabi and Mezic*, 2017; *Pan and Duraisamy*, 2020b; *Brunton et al.*, 2017; *Arbabi and Mezić*, 2017; *Mezić*, 2005; *Röjsel*, 2017). However, if one is interested in the strongly non-linear transient dynamics leading to an attractor or reduced-order modeling for high fidelity numerical simulations (*Xu and Duraisamy*, 2019; *Xu et al.*, 2020; *Huang et al.*, 2018; *Parish et al.*, 2020; *Carlberg et al.*, 2013), time-delay embedding may become less appropriate as several delay snapshots of the full order model are required to initialize the model. In that case, non-linear observables may be more appropriate.

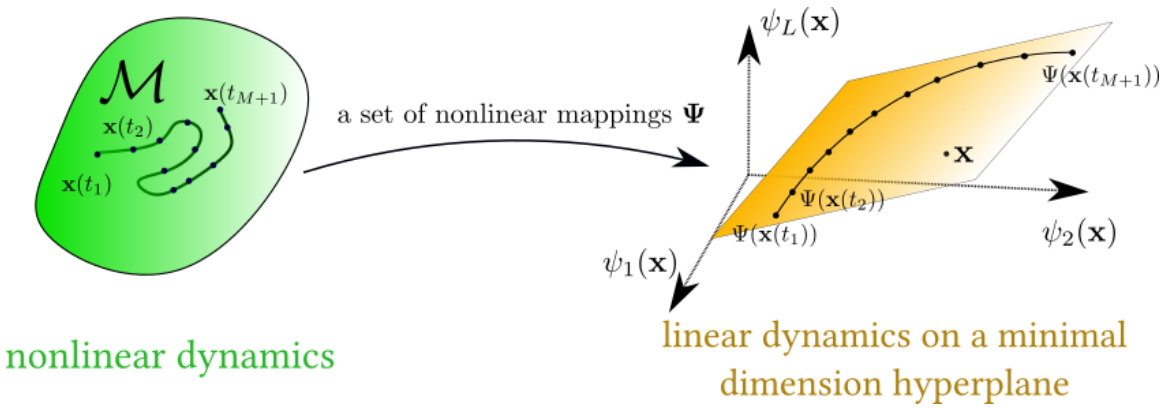


Figure 4.1: Sketch on finding the minimal Koopman invariant subspace.

Driven by the interest in modal analysis and control of transient flow phenomena, we consider augmentation of the observable space with non-linear functions of the state, e.g., EDMD (*Williams et al.*, 2015)/KDMD (*Williams et al.*, 2014). Although it has been reported that KDMD allows for a set of more interpretable Koopman eigenvalues (*Williams et al.*, 2014) and better accuracy (*Röjsel*, 2017), issues such as modes selection, spurious modes (*Kaiser et al.*, 2017; *Zhang et al.*, 2017), and choice of dictionary/kernel in EDMD/KDMD remain open. In fact, the choice of kernel type and hyperparameter can significantly affect the resulting eigenmodes, distribution of eigenvalues (*Kutz et al.*, 2016), and the accuracy of predictions (*Zhang et al.*, 2017).

Naturally, for a given observable f, we are often interested in the *minimal* $\mathcal{F}_D$ as shown in fig. 4.1. Searching for an accurate and informative Koopman-invariant subspace has long been a pursuit in the DMD community. *Rowley et al.* (2009) and *Schmid et al.* (2012) considered selecting dominant DMD modes in order of their amplitudes. However, following such a criterion (*Tu et al.*, 2013; *Kou and Zhang*, 2017), may result in the selection of unimportant modes that may have large amplitudes but decay rapidly. As a result, *Tu et al.* (2013) considered weighting the loss term by the magnitude of eigenvalues to penalize the retention of fast decaying modes. Sparsity-promoting DMD (referred as "spDMD" throughout the paper) developed by *Jovanović et al.* (2014) recasts mode selection in DMD as an optimization prob-

lem with a ℓ_1 penalty. With a preference of stable modes over fast decaying ones, *Tissot et al.* (2014) proposed a simpler criterion based on time-averaged-eigenvalue-weighted amplitude. This was followed by *Kou and Zhang* (2017) who used a similar criterion but computed the "energy" of each mode, yielding similar performance to spDMD at a lower computational cost. Based on the orthonormal property of pseudo-inverse, *Hua et al.* (2017) proposed an ordering of Koopman modes by defining a new "energy". Compared with previous empirical criteria, the "energy" for each mode involves a pseudo-inverse which combines the influence from all eigenmodes. Therefore the contribution from each mode cannot be isolated.

Instead of selecting modes from a "reconstruction" perspective, *Zhang et al.* (2017) studied the issue of spurious modes by evaluating the deviation of the identified eigenfunctions from linear evolution in an a priori sense. Further, Optimized DMD (*Chen et al.*, 2012; *Askham and Kutz*, 2018) combines DMD with mode selection simultaneously, which is the forerunner of recently proposed neural network-based models for Koopman eigenfunctions in spirit (*Takeishi et al.*, 2017; *Lusch et al.*, 2018b; *Otto and Rowley*, 2019b; *Pan and Duraisamy*, 2020a; *Li et al.*, 2017; *Yeung et al.*, 2019). Regardless of the above issues related to non-convex optimization (*Dawson et al.*, 2016), an extension of optimized DMD to EDMD/KDMD is not straightforward. Further, neural network-based models require large amounts of data, are prone to local minima, and lack interpretability.

There have been a few attempts towards mode selection in EDMD/KDMD. *Brunton et al.* (2016a) present an iterative method that augments the dictionary of EDMD until a convergence criterion is reached for the subspace. This is effectively a recursive implementation of EDMD. Recently, *Haseli and Cortés* (2019) showed that given a sufficient amount of data, if there is any accurate Koopman eigenfunction spanned by the dictionary, it must correspond to one of the resulting eigenvectors. Moreover, they proposed the idea of mode selection by checking if the reciprocal of identified

eigenvalue also appears when the temporal sequence of data is reversed, which is similar to comparing eigenvalues on the complex plane from different trajectories, as proposed by *Hua et al.* (2017). In contrast to the "bottom-up" method of *Brunton et al.* (2016a) with subspace augmentation, *Hua et al.* (2017) proposed a "top-down" subspace subtraction method relying on iteratively projecting the features onto the null space. A similar idea can be traced back to *Kaiser et al.* (2017) who propose a search for the sparsest vector in the null space.

In this chapter, we are interested in learning an *accurate minimal* Koopman-invariant subspace for the full state $\mathbf{x}$. As illustrated in fig. 4.1, extracting such a subspace yields the linear representation we seek and results in useful coordinates for a multitude of applications, including modal analysis and control (*Arbabi et al.*, 2018a). As the main contribution of this work, we propose a novel sparsity promoting framework equipped with the following strategy to extract an accurate yet minimal Koopman-invariant subspace:

1. We first evaluate the normalized maximal deviation of the evolution of each eigenfunction from linear evolution in a posteriori fashion.

2. Using the above criteria, we select a user-defined number of accurate EDMD/KDMD modes;

3. Among the accurate EDMD/KDMD modes obtained above, informative modes are selected using multi-task feature learning (*Argyriou et al.*, 2008a; *Bach et al.*, 2012).

4.2 Challenges

Recall that we have briefly intorduced EDMD and KDMD section 2.2 and section 2.3. In this section, we will further discuss two existing challenges in the use of EDMD and KDMD for Koopman analysis.

4.2.1 Mode Selection

The number of approximated Koopman tuples (eigenfunction, eigenvalue, modes) from EDMD grows with the dictionary size, whereas the KDMD grows with the number of snapshots. However, in most cases, a significant number of the eigenfunctions fail to evolve linearly, or are redundant in contribution to the reconstruction of the state $\mathbf{x}$. For example as shown by *Budišić et al.* (2012), the Koopman eigenfunctions that vanish nowhere form an Abelian group under pointwise products of functions, while polynomial observables evolve linearly for a general linear system. These eigenfunctions, associated with the polynomial observables, are redundant in terms of providing an intrinsic coordinate for the linear dynamics.

When the number of features is larger than the number of data snapshots, EDMD eigenvalues can be misleading (*Otto and Rowley*, 2019b) and often plagued with spurious eigenfunctions that do not evolve linearly even when the number of data snapshots is sufficient. Analytically, it is clear that a Koopman eigenfunction in the span of the dictionary will be associated with one of the eigenvectors obtained from EDMD, given $\mathbf{\Psi_x}$ is full rank, and contains sufficient snapshots M (*Haseli and Cortés*, 2019). Indeed, the EDMD is a L_2 projection of the Koopman operator under the empirical measure (*Korda and Mezić*, 2018a). As a result, we seek a Koopman-invariant subspace following the standard EDMD/KDMD. Since KDMD can be viewed as an efficient way of populating a dictionary of nonlinear features in high dimensional spaces, the above arguments apply to KDMD as well. It should be noted that numerical conditioning can play a critical role since full rank matrices can be ill-conditioned.

4.2.2 Choice of Hyperparameters in Dictionary-based Methods

Although the use of a kernel defines an infinite-dimensional feature space, the resulting finite number of effective features can still be affected by both the type of

the kernel and the hyperparameters in the kernel as clearly shown by *Kutz et al.* (2016). Compared to EDMD/KDMD, which are based on a fixed dictionary of features, neural network approaches (*Otto and Rowley*, 2019b; *Pan and Duraisamy*, 2020a; *Lusch et al.*, 2018b) have the potential to be more expressive in searching for a larger Koopman-invariant subspace. From a kernel viewpoint (*Cho and Saul*, 2009), feedforward neural networks enable adaptation of the kernel function to the data. Such a characteristic could become significant when the underlying Koopman eigenfunction is discontinuous. From an efficiency standpoint, a kernel-guided scalable EDMD (*DeGennaro and Urban*, 2019) may be pursued. This can be achieved by generating kernel-consistent random Fourier features or approximating a few components of the feature vector constructed from Mercer's theorem, i.e., the eigenfunctions of the Hilbert–Schmidt integral operator on the RKHS.

4.3 Methodology & Analysis

To address the challenges described in section 4.2, we develop a novel framework that uses EDMD/KDMD modes to identify a sparse, accurate, and informative Koopman-invariant subspace. Our framework first prunes spurious, inaccurate eigenmodes and second determines a sparse representation of the system state $\mathbf{x}$ from the accurate eigenmodes. In addition to the training data, as required in standard EDMD/KDMD, a validation trajectory data-set is required to avoid overfitting on training data. The terms spEDMD/spKDMD will refer to filtered mode selections of EDMD and KDMD, respectively.

4.3.1 Pruning spurious modes by a posteriori error analysis

Given a validation trajectory $\mathbf{x}(t)$ where $t \in [0, T]$ associated with the non-linear dynamical system, for $i = 1, \ldots, L$, we define the goodness of i-th eigenfunctions in a posteriori way as the maximal normalized deviation from linear evolution conditioned

on trajectory $\mathbf{x}(t)$ as Q_i in the form

$$e_{i,\mathbf{x}(0)}(t) = \frac{|\varphi_i(\mathbf{x}(t)) - e^{\lambda_i t}\varphi_i(\mathbf{x}(0))|}{\|\varphi_i(\mathbf{x})\|_2}, \tag{4.1}$$

$$Q_i \triangleq e^{max}_{i,\mathbf{x}(0)} = \max_t e_{i,\mathbf{x}(0)}(t), \tag{4.2}$$

where $\|\varphi_i(\mathbf{x})\|_2 \triangleq \sqrt{\frac{1}{T}\int_0^T \varphi_i^*(\mathbf{x}(t))\varphi_i(\mathbf{x}(t))dt}$. In practice, we evaluate the above integral terms discretely in time. A similar a priori and less restrictive method has been previously proposed (*Zhang et al.*, 2017). In contrast, in the proposed method, the maximal error is evaluated in an a posteriori way to better differentiate spurious modes from accurate ones. For any $1 \leqslant \hat{L} \leqslant L$, we can always select top $\hat{L}$ accurate eigenmodes out of L eigenmodes denoting their index in eigen-decomposition as $\{i_1, i_2, \ldots, i_{\hat{L}}\}$, i.e., $Q_{i_1} \leqslant \ldots \leqslant Q_{i_{\hat{L}}} \leqslant \ldots \leqslant Q_{i_L}$. Then, for the next sparse reconstruction step, we simply use $\boldsymbol{\varphi}$ defined as follows to reconstruct the state $\mathbf{x}$,

$$\boldsymbol{\varphi}_{\hat{L}}(\mathbf{x}(t)) \triangleq \begin{bmatrix} \varphi_{i_1}(\mathbf{x}(t)) & \ldots & \varphi_{i_{\hat{L}}}(\mathbf{x}(t)) \end{bmatrix} \in \mathbb{C}^{\hat{L}}. \tag{4.3}$$

To choose an appropriate $\hat{L}$ to linearly reconstruct the system state $\mathbf{x}$, we monitor the normalized reconstruction error for the aforementioned set of top $\hat{L}$ accurate eigenmodes in the following form

$$R_{\hat{L}} \triangleq \frac{\|(\mathbf{I} - \boldsymbol{\Psi}_{\hat{L}}\boldsymbol{\Psi}^+_{\hat{L}})\mathbf{X}\|_F}{\|\mathbf{X}\|_F}, \tag{4.4}$$

where $\mathbf{I}$ is the identity matrix, and

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_{M-1} \end{bmatrix}, \quad \boldsymbol{\Psi}_{\hat{L}} = \begin{bmatrix} \boldsymbol{\varphi}_{\hat{L}}(\mathbf{x}_0) \\ \vdots \\ \boldsymbol{\varphi}_{\hat{L}}(\mathbf{x}_{M-1}) \end{bmatrix}. \tag{4.5}$$

As a result, the evaluation of eq. (4.4) for each $\hat{L}$ is of similar expense to least-square regression. For an increasing number of selected eigenfunctions $\hat{L}$, the reconstruction error $R_{\hat{L}}$ decreases, while the largest linear evolution error $Q_{i_{\hat{L}}}$ increases. Then, a truncation $\hat{L}$ can be defined by the user to strike a balance between linear evolution error $Q_{i_{\hat{L}}}$ and reconstruction error $R_{\hat{L}}$. In the next subsection, we will further select a subset of eigenmodes for spanning the minimal Koopman-invariant subspace.

4.3.2 Sparse reconstruction via multi-task feature learning

Numerical experiments revealed that, in the selected set of $\hat{L}$ most accurate eigenfunctions, two types of redundant eigenfunctions were found:

1. Nearly constant eigenfunctions with eigenvalues close to zero,

2. Pointwise products of Koopman eigenfunctions introduced by non-linear observables, not useful in linear reconstruction.

To filter the above modes, we consider sparse regression with $\hat{L}$ most accurate eigenfunctions as features and the system state $\mathbf{x}$ as target. Note that, since we have guaranteed the accuracy of selected eigenmodes, one can either choose features a priori: $\varphi_i(\mathbf{x}(t))$ or a posteriori (multi-step prediction) $e^{\lambda_i t}\varphi_i(\mathbf{x}(0))$. Here we choose the latter since it is directly related to prediction, and can actually be reused from the previous step without additional computational cost. We denote the corresponding multi-step prediction feature matrix as $\hat{\mathbf{\Psi}}_{\hat{L}}$,

$$\hat{\mathbf{\Psi}}_{\hat{L}} = \begin{bmatrix} \boldsymbol{\varphi}_{\hat{L}}(\mathbf{x}_0) \\ \boldsymbol{\varphi}_{\hat{L}}(\mathbf{x}_0)e^{\Delta t \Lambda_{\hat{L}}} \\ \vdots \\ \boldsymbol{\varphi}_{\hat{L}}(\mathbf{x}_0)e^{(M-1)\Delta t \Lambda_{\hat{L}}} \end{bmatrix} \in \mathbb{C}^{M \times \hat{L}}, \tag{4.6}$$

where $\Lambda_{\hat{L}} = \mathrm{diag}(\lambda_{i_1}, \ldots, \lambda_{i_{\hat{L}}})$. Note that similar features $\hat{\mathbf{\Psi}}_{\hat{L}}$ were also considered in

sparsity-promoting DMD (*Jovanović et al.*, 2014) and optimized DMD (*Chen et al.*, 2012). Finally, the fact that there is no control over the magnitudes of the implicitly defined features in the standard KDMD may cause unequal weighting between different features. Thus, we consider scaling the initial value of all eigenfunctions to be unity in eq. (4.7),

$$\hat{\mathbf{\Psi}}_{\hat{L},\text{scaled}} = \hat{\mathbf{\Psi}}_{\hat{L}} \mathbf{\Lambda}_{ini}^{-1} = \begin{bmatrix} 1 & \dots & 1 \\ e^{\Delta t \lambda_{i_1}} & \dots & e^{\Delta t \lambda_{i_{\hat{L}}}} \\ \vdots & \vdots & \vdots \\ e^{(M-1)\Delta t \lambda_{i_1}} & \dots & e^{(M-1)\Delta t \lambda_{i_{\hat{L}}}} \end{bmatrix}, \tag{4.7}$$

where

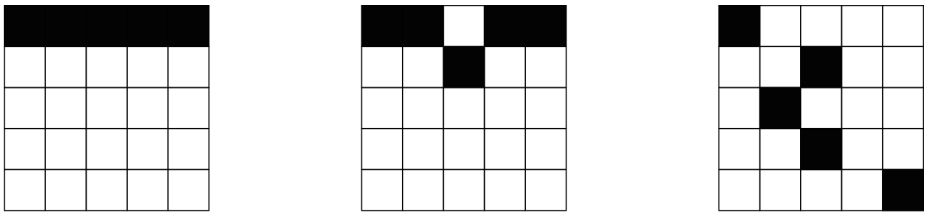
$$\mathbf{\Lambda}_{ini} = \text{diag}(\begin{bmatrix} \varphi_{i_1}(\mathbf{x}_0) & \dots & \varphi_{i_{\hat{L}}}(\mathbf{x}_0) \end{bmatrix}). \tag{4.8}$$

Since $\mathbf{x}$ is finite-dimensional, searching for a sparse combination of $\hat{\mathbf{\Psi}}_{\hat{L}}$ to reconstruct $\mathbf{x}$ is *equivalent* to the solution of a multi-task feature learning problem with preference over a relatively small size of features. Note that this type of problem has been studied extensively in the machine learning community (*Argyriou et al.*, 2008a; *Zhao et al.*, 2015; *Argyriou et al.*, 2008b). In this work, given $\mathbf{X}$ and $\hat{\mathbf{\Psi}}_{\hat{L},\text{scaled}}$, we leverage the multi-task ElasticNet (*Pedregosa et al.*, 2011) to search for a row-wise sparse $\mathbf{B}'$, which solves the following convex optimization problem:

$$\mathbf{B}'^{*} = \underset{\mathbf{B}' \in \mathbb{C}^{\hat{L} \times n}}{\arg\min} \frac{1}{2M} \|\mathbf{X} - \hat{\mathbf{\Psi}}_{\hat{L},\text{scaled}} \mathbf{B}'\|_F^2 + \alpha\rho \|\mathbf{B}'\|_{2,1} + \frac{\alpha(1-\rho)}{2} \|\mathbf{B}'\|_F^2, \tag{4.9}$$

and

$$\mathbf{B} = \mathbf{\Lambda}_{ini}^{-1} \mathbf{B}'^{*}, \tag{4.10}$$



ℓ_1/ℓ_2 loss $= \sqrt{N}$ ℓ_1/ℓ_2 loss $= 1 + \sqrt{N-1}$ ℓ_1/ℓ_2 loss $= N$

Figure 4.2: Illustration of ℓ_1/ℓ_2 norm (defined in eq. (4.11)) for different $N \times N$ 0-1 binary matrices.

where $\|\cdot\|_{2,1}$ defined in eq. (4.11) is the so-called ℓ_1/ℓ_2 norm for a matrix $\mathbf{W}$,

$$\|\mathbf{W}\|_{2,1} \triangleq \sum_i \sqrt{\sum_j \mathbf{W}_{ij}^2} = \sum_i \|\mathbf{w}_i\|_2, \tag{4.11}$$

and $\mathbf{W}_i$ is i-th row of $\mathbf{W}$. This norm is special in that it controls the number of shared features learned across all tasks, i.e., i-th Koopman mode $\mathbf{b}_i$ is either driven to a zero vector or not while the standard ℓ_1 only controls the number of features for each task independently.

As a simple illustration, the ℓ_1/ℓ_2 norm for three different $N \times N$ square matrices (here $N = 5$) with 0-1 binary entries is displayed in fig. 4.2. Since $\sqrt{N} \leqslant 1+\sqrt{N-1} \leqslant N$, minimizing the ℓ_1/ℓ_2 norm leads to a penalty on the number of rows. As shown in the second term on the right hand side of eq. (4.9), minimizing the ℓ_1/ℓ_2 norm penalizes the number of Koopman eigenmodes.

The above procedure not only serves the purpose of selecting modes that explains the behavior of all components in the state, but is also particularly natural for EDMD/KDMD since Koopman modes are obtained via regression. α is a penalty coefficient that controls the amount of total regularization in the ℓ_1/ℓ_2 and ℓ_2 norms, while ρ is the ElasticNet mixing parameter (*Zou and Hastie*, 2005) that ensures uniqueness of the solution when highly correlated features exist. In our case, we choose $\rho = 0.99$ and sweep α over a certain range with L_r non-zero features denoted

as $\hat{\mathbf{\Psi}}_{L_r}$ for each α, while monitoring the normalized residual $\min_{\mathbf{B}}\|\mathbf{X}-\hat{\mathbf{\Psi}}_{L_r}\mathbf{B}\|_F/\|\mathbf{X}\|_F$ to choose a proper α. It has to be mentioned that, sometimes the sparsest solution from a multi-task ElasticNet was found to shrink to a small number instead of zero. This is a consequence of the insufficiency of the current optimization algorithm which employs coordinate descent (*Pedregosa et al.*, 2011). Hence for each target component, we consider an additional hard-thresholding step by setting the corresponding magnitude of the coefficient, i.e., contribution of any mode, to zero if it is smaller than a certain threshold $\epsilon \in [10^{-2}, 10^{-3}]$.

Finally, we refit the Koopman modes as $\mathbf{B}_{L_r} = \hat{\mathbf{\Psi}}^{+}_{L_r}\mathbf{X}$ which avoids the bias introduced by the penalty term [1] in eq. (4.9). To summarize, the general idea of the framework is illustrated in fig. 4.3. As a side note for interested readers, if one only performs multi-task feature learning without hard-thresholding and refitting, one would obtain a smooth ElasticNet path instead of a discontinuous one with hard-thresholding and refitting. However, the smooth ElasticNet can lead to difficulties in choosing the proper α visually, especially when the given dictionary of EDMD/KDMD is not rich enough to cover an informative Koopman-invariant subspace. Further discussion on the computational complexity of our framework is presented in appendix C.

Thus far, we have presented our main contribution: a novel optimization-based framework to search for an accurate and minimal Koopman-invariant subspace from data. An appealing aspect of our framework is the *model agnostic property*, which makes the extension easy from the standard EDMD/KDMD to more advanced approximation methods (*Jungers and Tabuada*, 2019; *Mamakoukas et al.*, 2019; *Azencot et al.*, 2019). In the following subsection, we present two mathematical insights: 1) multi-task feature learning generalizes spDMD under a specific constraint; 2) a popular empirical criterion can be viewed as a single step of proximal gradient descent.

[1] spDMD does not refit $\mathbf{B}$

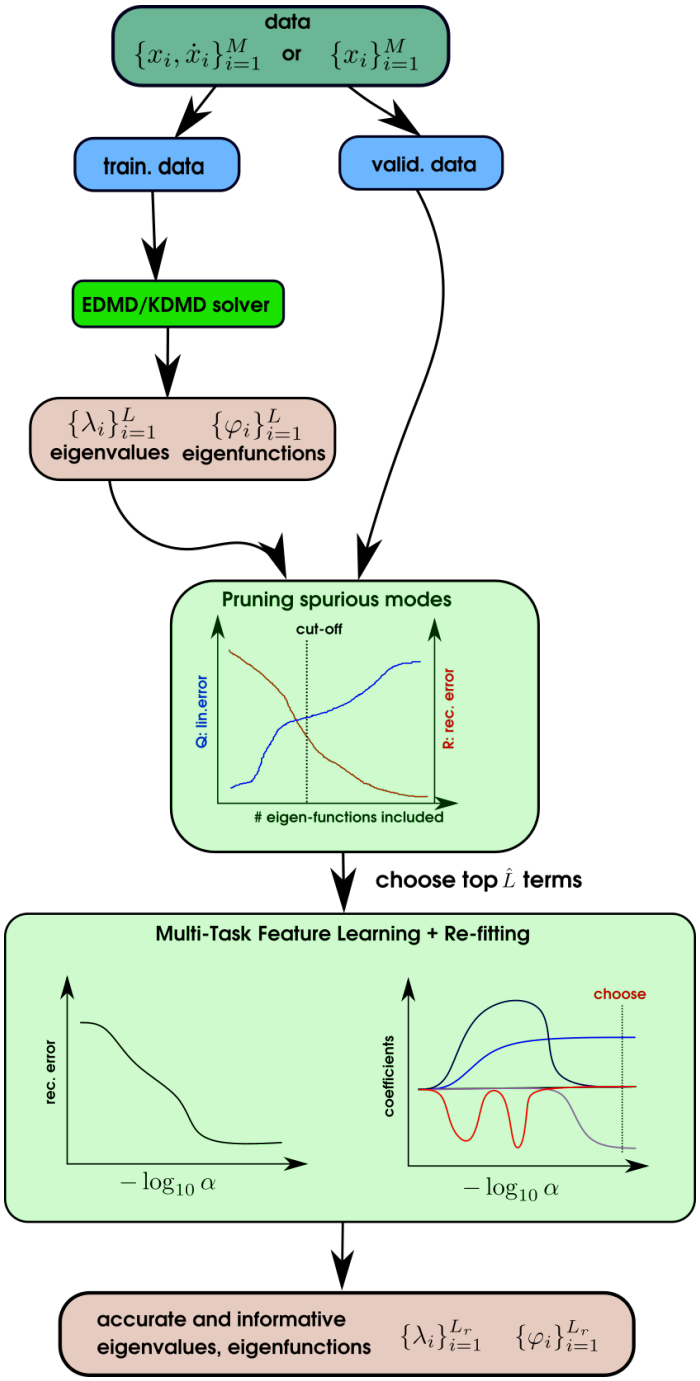


Figure 4.3: Schematic illustrating the idea of sparse identification of Koopman-invariant subspaces for EDMD and KDMD.

4.3.3 Relationship between sparsity-promoting DMD, Kou's criterion and multi-task feature learning

For simplicity, neglecting the ElasticNet part (i.e., using $\rho = 1$), eq. (4.9) with L modes leads to a multi-task Lasso problem,

$$\min_{\mathbf{B}' \in \mathbb{C}^{L \times n}} \frac{1}{2M} \|\mathbf{X} - \hat{\mathbf{\Psi}}_{L,\text{scaled}} \mathbf{B}'\|_F^2 + \alpha \|\mathbf{B}'\|_{2,1}. \tag{4.12}$$

Recall that in spDMD (*Jovanović et al.*, 2014), DMD modes $\phi_1, \ldots, \phi_L$ with $\|\phi_i\|_2 = 1$ remain the same as standard DMD. Similariy, if we posit a structural constraint on $\mathbf{B}'$ in eq. (4.12) by enforcing the modes as those from DMD, then there exist $\alpha_1, \ldots, \alpha_L$ such that,

$$\mathbf{B}' = \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_L \end{bmatrix} \begin{bmatrix} \phi_1^{\mathrm{T}} \\ \vdots \\ \phi_L^{\mathrm{T}} \end{bmatrix}. \tag{4.13}$$

Note the fact that $\|\mathbf{B}'\|_{2,1} = \sum_i^L |\alpha_i|$. Hence, we recover the ℓ_1 optimization step in the spDMD (*Jovanović et al.*, 2014) from eq. (4.12) as,

$$\min_{\alpha_1,\ldots,\alpha_L \in \mathbb{C}} \frac{1}{2M} \left\| \mathbf{X} - \begin{bmatrix} 1 & \ldots & 1 \\ \vdots & \vdots & \vdots \\ e^{(M-1)\Delta t \lambda_{i_1}} & \ldots & e^{(M-1)\Delta t \lambda_{i_L}} \end{bmatrix} \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_L \end{bmatrix} \begin{bmatrix} \phi_1^{\mathrm{T}} \\ \vdots \\ \phi_L^{\mathrm{T}} \end{bmatrix} \right\|_F^2 + \alpha \sum_{i=1}^{L} |\alpha_i|, \tag{4.14}$$

where $\phi_1, \ldots, \phi_L$ are the DMD modes corresponding to eigenvalues as $\lambda_1, \ldots, \lambda_L$. Hence, multi-task feature learning solves a less constrained optimization than spDMD in the context of DMD.

Kou and Zhang (2017) proposed an empirical criterion for mode selection by ordering modes with "energy" I_i defined as

$$I_i = \sum_{j=1}^{M} |\alpha_i e^{(j-1)\Delta t \lambda_i}| = \begin{cases} \frac{|\alpha_i|(1-|e^{\Delta t \lambda_i}|^M)}{1-|e^{\Delta t \lambda_i}|}, & \text{if } |e^{\Delta t \lambda_i}| \neq 1, \\ M|\alpha_i|, & \text{otherwise,} \end{cases} . \tag{4.15}$$

From an optimization viewpoint, consider a posteriori prediction matrix $\mathbf{X}_{\text{DMD}}$ from DMD

$$\mathbf{X} \approx \mathbf{X}_{\text{DMD}} = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \vdots & \vdots \\ e^{(M-1)\Delta t \lambda_1} & \dots & e^{(M-1)\Delta t \lambda_L} \end{bmatrix} \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_L \end{bmatrix} \begin{bmatrix} \phi_1^{\mathrm{T}} \\ \vdots \\ \phi_L^{\mathrm{T}} \end{bmatrix}, \tag{4.16}$$

where $\mathbf{X}_{\text{DMD}}$ is determined from DMD using the snapshot pair $(\mathbf{X}, \mathbf{X}')$. $\mathbf{X}_{\text{DMD}}$ is a rank-1 summation of contributions from different modes (*Schmid*, 2010). Hence, a general mode selection technique with a user-defined preference weighting $\mathbf{w}$ is the following weighted ℓ_0 nonconvex optimization problem:

$$\min_{\mathbf{a} \in \mathbb{C}^L} \left\| \mathbf{X}_{\text{DMD}} - \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \vdots & \vdots \\ e^{(M-1)\Delta t \lambda_1} & \dots & e^{(M-1)\Delta t \lambda_L} \end{bmatrix} \text{diag}(\mathbf{a}) \begin{bmatrix} \phi_1^{\mathrm{T}} \\ \vdots \\ \phi_L^{\mathrm{T}} \end{bmatrix} \right\|_F^2 + \lambda \|\mathbf{a}\|_{\mathbf{w},0} \tag{4.17}$$

where $\|\mathbf{a}\|_{\mathbf{w},0} \triangleq \sum_i w_i |a_i|^0$, $|a_i|^0$ is one if $a_i \neq 0$ and zero otherwise. Note that this pseudo-norm can be viewed as a limiting case of a weighted composite sine function smoothed ℓ_0 regularization (*Wang et al.*, 2019).

To solve this non-convex optimization problem, compared to the popular ℓ_1 relaxation method such as the one in sparsity-promoting DMD, a less-known but rather efficient way is iterative least-squares hard thresholding. This has been used in sparse identification of dynamical systems (SINDy) (*Brunton et al.*, 2016b), and conver-

gence to a local minimum has been proved (*Zhang and Schaeffer*, 2019). Indeed, a more rigorous framework that generalizes such an algorithm is the proximal gradient method (*Parikh et al.*, 2014). Much like Newton's method is a standard tool for unconstrained smooth optimization, the proximal gradient method is the standard tool for constrained non-smooth optimization. Here, it is straightforward to derive the iterative algorithm that extends to the weighted ℓ_0 norm from step k to step $k+1$ as

$$\mathbf{a}^{k+1} = \text{prox}_{\frac{\lambda}{2}\eta_k\|\cdot\|_{\mathbf{w},0}}(\mathbf{a}^k - \eta_k \nabla_{\mathbf{a}} \mathcal{Q}(\mathbf{a}^k)), \tag{4.18}$$

where

$$\mathcal{Q}(\mathbf{a}) = \frac{1}{2}\left\| \mathbf{X}_{\text{DMD}} - \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \vdots & \vdots \\ e^{(M-1)\Delta t \lambda_1} & \dots & e^{(M-1)\Delta t \lambda_L} \end{bmatrix} \text{diag}(\mathbf{a}) \begin{bmatrix} \phi_1^{\text{T}} \\ \vdots \\ \phi_L^{\text{T}} \end{bmatrix} \right\|_F^2, \tag{4.19}$$

and η_k is the step-size at step k. Notice that the weighted ℓ_0 norm is a separable sum of a_i. After some algebra, we have the proximal operator as

$$\text{prox}_{\frac{\lambda}{2}\eta_k\|\cdot\|_{\mathbf{w},0}}(\mathbf{a}) = \begin{bmatrix} \mathcal{H}_{\sqrt{\lambda\eta_k}}(a_1/\sqrt{w_1}) & \dots & \mathcal{H}_{\sqrt{\lambda\eta_k}}(a_L/\sqrt{w_L}) \end{bmatrix}^{\text{T}}, \tag{4.20}$$

where $\mathcal{H}_{\sqrt{\lambda\eta_k}}(a)$ is an element-wise hard thresholding operator defined as a if $|a| < \sqrt{\lambda\eta_k}$ and zero otherwise.

Particularly, if one considers the initial step-size to be extremely small $\eta_1 \ll 1$, then the second term in eq. (4.18) can be neglected. Thus, for $i = 1, \dots, L$, with the following weighting scheme that penalizes fast decaying modes:

$$w_i = 1/\beta_i^2, \quad \beta_i = \begin{cases} \frac{1-|e^{\Delta t \lambda_i}|^M}{1-|e^{\Delta t \lambda_i}|}, & \text{if } |e^{\Delta t \lambda_i}| \neq 1, \\ M, & \text{otherwise,} \end{cases} \tag{4.21}$$

one immediately realizes the thresholding criterion for i-th entry of $\mathbf{a}$ becomes

$$\sqrt{\lambda\eta_k} > |\alpha_i/\sqrt{w_i}| = |\alpha_i\beta_i|. \tag{4.22}$$

Then plugging eq. (4.21) in eq. (4.18), the first iteration in eq. (4.18) reduces to mode selection with Kou's criterion in eq. (4.15). Normally, β_i is very large for unstable modes and small for decaying modes. It is important to note that a) such a choice of $\mathbf{w}$ preferring unstable/long-lasting modes over decaying modes is still user-defined; 2) Optimization is in an a priori sense to obtain DMD. Thus, the insufficiency of the a priori formulation to account for temporal evolution is indeed *compensated* by this criterion, while DMD in an a posteriori formulation (e.g., sparsity-promoting DMD) includes such a effect implicitly in the optimization. Hence, it is possible that in some circumstances spDMD and Kou's criterion could achieve similar performance (*Kou and Zhang*, 2017).

Lastly, as summarized in fig. 4.4, it is important to mention the similarities and differences between spKDMD and spDMD: 1) spKDMD will refit Koopman modes while spDMD does not; and 2) The amplitude for all the modes in spKDMD is fixed as unity while it has to be determined in spDMD.

4.3.4 Hyper-parameter selection

For simplicity, hyper-parameter selection for KDMD is only discussed in this section. To fully determine a kernel in KDMD, one would have to choose the following:

1. kernel type,
2. kernel parameters, e.g., scale parameters σ,
3. rank truncation r.

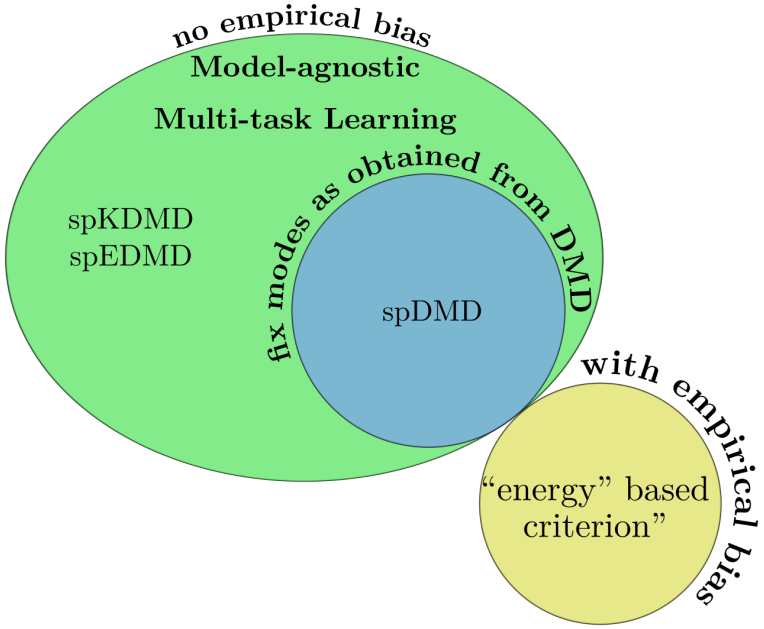


Figure 4.4: Differences and similarities among existing mode selection methods.

In this work, for simplicity, we fix the kernel type to be an isotropic Gaussian. Motivated by previous work on error evaluation in Koopman modes by *Zhang et al.* (2017), we consider evaluation with cross validation on a priori mean normalized accuracy defined in eqs. (4.23) and (4.24) for i-th eigenfunction,

$$\text{discrete form: } Q_i^a = \frac{1}{M-1}\sum_{j=0}^{M-2}\frac{|\varphi_i(\mathbf{x}_{j+1}) - \lambda_i\varphi_i(\mathbf{x}_j)|}{\sqrt{\frac{1}{M}\sum_{k=0}^{M-1}\varphi_i^*(\mathbf{x}_k)\varphi_i(\mathbf{x}_k)}} \tag{4.23}$$

$$\text{continuous form: } Q_i^a = \frac{1}{M}\sum_{j=0}^{M-1}\frac{|\dot{\mathbf{x}}_j\cdot\nabla_{\mathbf{x}}\varphi_i(\mathbf{x}_j) - \lambda_i\varphi_i(\mathbf{x}_j)|}{\sqrt{\frac{1}{M}\sum_{k=0}^{M-1}\varphi_i^*(\mathbf{x}_k)\varphi_i(\mathbf{x}_k)}} \tag{4.24}$$

on validation data for different number of rank truncation and kernel parameters.

Note that evaluation on maximal instead of mean normalized accuracy would lead to the error metric to be strongly dependent on the local sparsity of training data in the feature space. This is particularly true for a single trajectory for a

high-dimensional dynamical system is used, and random shuffled cross validation is performed (*Pan and Duraisamy*, 2018c).

For each set of hyperparameters, we first compute the number of eigenfunctions of which the error defined in eqs. (4.23) and (4.24) is below a certain threshold on both training and validation data for each fold of cross validation. Then we compute the average number of such eigenfunctions over all folds which indicates the quality of the corresponding subspace. Finally, we plot the average number versus rank truncation r and kernel scale parameters σ to select hyperparameters.

4.3.5 Implementation

We implement the described framework in Python with moderate parallelism in each module. We use `scipy.special.hermitenorm` (*Jones et al.*, 2001–) to generate normalized Hermite polynomials and `MultiTaskElasticNet` in the `scikit-learn` (*Pedregosa et al.*, 2011) for multi-task feature learning where we set the maximal iteration as 10^5 and tolerance as 10^{-12}. MPI parallelism using `mpi4py` (*Dalcin et al.*, 2011) is used for the grid search in hyperparameter selection. To prepare data with hundreds of gigabytes collected from high fidelity simulations, a distributed SVD written in C++ named Parallel Data Processing (PDP) Tool is developed for dimension reduction (*Arnold-Medabalimi et al.*, 2020).

4.4 Applications

4.4.1 Simple Non-linear ODE System

We first consider a simple fixed point non-linear dynamical system which has an analytically determined, finite-dimensional non-trivial Koopman-invariant subspace (*Brunton et al.*, 2016a; *Kaiser et al.*, 2017) to show the effectiveness of proposed method. We consider a continuous-time formulation. The governing equation for the

dynamical system is given as follows,

$$\dot{x}_1 = \mu x_1, \tag{4.25}$$

$$\dot{x}_2 = \lambda(x_2 - x_1^2), \tag{4.26}$$

where $\mu = -0.05, \lambda = -1$. One natural choice of the minimal Koopman eigenfunctions that *linearly* reconstructs the state is (*Brunton et al.*, 2016a)

$$\varphi_1(\mathbf{x}) = x_2 - \lambda x_1^2/(\lambda - 2\mu), \quad \varphi_2(\mathbf{x}) = x_1, \quad \varphi_3(\mathbf{x}) = x_1^2 \tag{4.27}$$

with eigenvalues $\lambda = -1$, $\mu = -0.05$, $2\mu = -0.1$ respectively.

The way we generate training, validation, and testing data is described below with distribution of the data shown in fig. 4.5,

1. *training data*: a point cloud with $M = 1600$ pairs of $\{\mathbf{x}^{(i)}, \dot{\mathbf{x}}^{(i)}\}_{i=0}^{M-1}$, is generated by Latin hypercube sampling (*Baudin et al.*, 2015) within the domain $x_1, x_2 \in [-0.5, 0.5]$.

2. *validation data*: a single trajectory with initial condition as $x_1(0) = x_2(0) = 0.4$, sampling time interval $\Delta t = 0.03754$ from $t = 0$ to $t = 30$.

3. *testing data*: a single trajectory with initial condition as $x_1(0) = x_2(0) = -0.3$, sampling time interval $\Delta t = 0.06677$ from $t = 0$ to $t = 40$.

As an illustration, we consider two models to approximate the Koopman operator from training data:

1. a continuous-time EDMD with normalized Hermite polynomials up to fifth order with $L = 36$ features,

2. a continuous-time KDMD with isotropic Gaussian kernel $\sigma = 2$ with reduced rank $r = L = 36$.

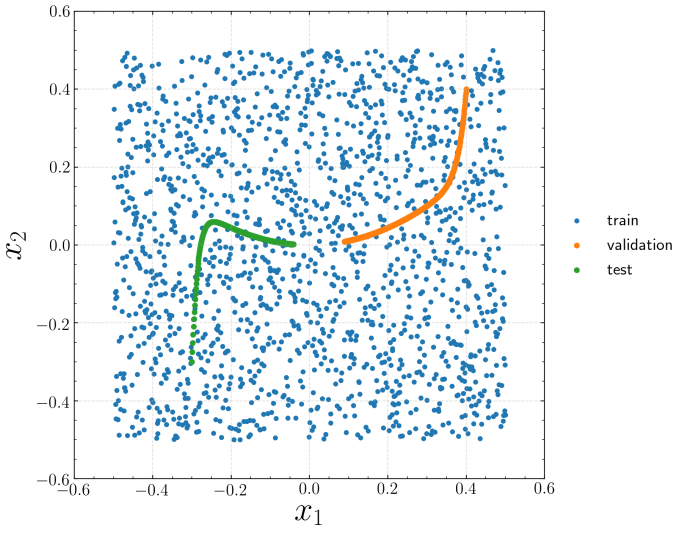


Figure 4.5: Data distribution for 2D fixed point attractor.

Details of the above choices based on the steps of hyperparameter selection in section 4.3.4 are given in appendix D.1.

4.4.1.1 Results for continuous-time EDMD with mode selection

As displayed in fig. 4.6, we begin with an error analysis of all of the eigenmodes on validation data in fig. 4.5 according to linearly evolving error Q defined in eq. (4.2) and R defined in eq. (4.4). From the left subfigure in fig. 4.6, considering both the linearly evolving error and the quality of the reconstruction, we choose the cut-off threshold at $\hat{L} = 10$. We observe a sharp cut-off in the left subfigure in fig. 4.6 around the number of selected eigenmodes $\hat{L} = 8$. This is a reasonable choice, since from the eigenvalues in the right subfigure in fig. 4.6, we notice the analytic Koopman eigenmodes are not covered until first 8 accurate eigenmodes are selected. Note that the legend in the right subfigure is ordered by the maximal deviation from linear evolution, e.g., the second most accurate mode is 34-th mode with zero eigenvalue. Indeed, the first four eigenfunctions (index=$1, 34, 35, 36$) are redundant in terms of reconstruction in this problem [2]. The fifth (index=29) and sixth (index=33) eigenmodes correspond to

[2]This could be interesting if the system is instead Hamiltonian.

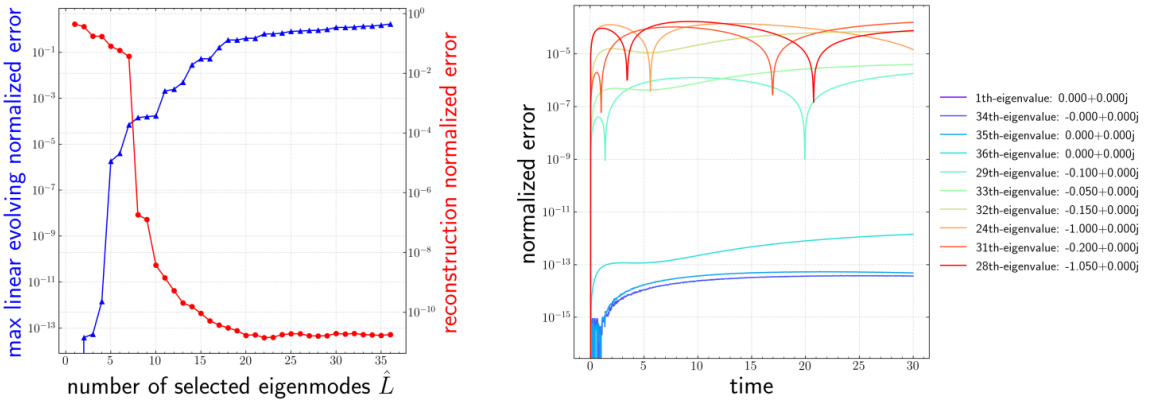


Figure 4.6: Error analysis of 36 eigenmodes from continuous-time EDMD for the 2D fixed point attractor. Left: trends of linearly evolving error Q and reconstruction error R. Right: temporal evolution of relative error for top $\hat{L} = 10$ accurate eigenmodes.

two of the analytic eigenfunctions that span the system, and the seventh (index=32) eigenmode is indeed the product of the fifth and sixth eigenfunctions. Similarly, the ninth and tenth eigenfunctions (index=31, 28) also appear to be the polynomial combination of the true eigenfunctions.

According to eq. (4.9), to further remove redundant modes, we perform multi-task feature learning on the $\hat{L} = 10$ eigenmodes . The corresponding ElasticNet path is shown in fig. 4.7. Note that each α corresponds to a minimizer of eq. (4.9). To choose a proper α so as to find a proper Koopman-invariant subspace, it is advisable to check the trend of the normalized reconstruction error and number of non-zero features. Given the dictionary, for simple problems for which there exists an *exact* Koopman-invariant subspace that also spans system state, a proper model can be obtained by selecting $\alpha \approx 10^{-6}$ which ends up with only 3 eigenfunctions as shown in fig. 4.7. Moreover, as is common for EDMD with polynomial basis (*Williams et al.*, 2014, 2015), a pyramid of eigenvalues appears in fig. 4.7.

As shown in fig. 4.8, both the identified eigenvalues, and contour of the phase angle and magnitude of selected eigenfunctions from spEDMD match the analytic eigenfunctions given in eq. (4.27) very well. As expected, the prediction on unseen

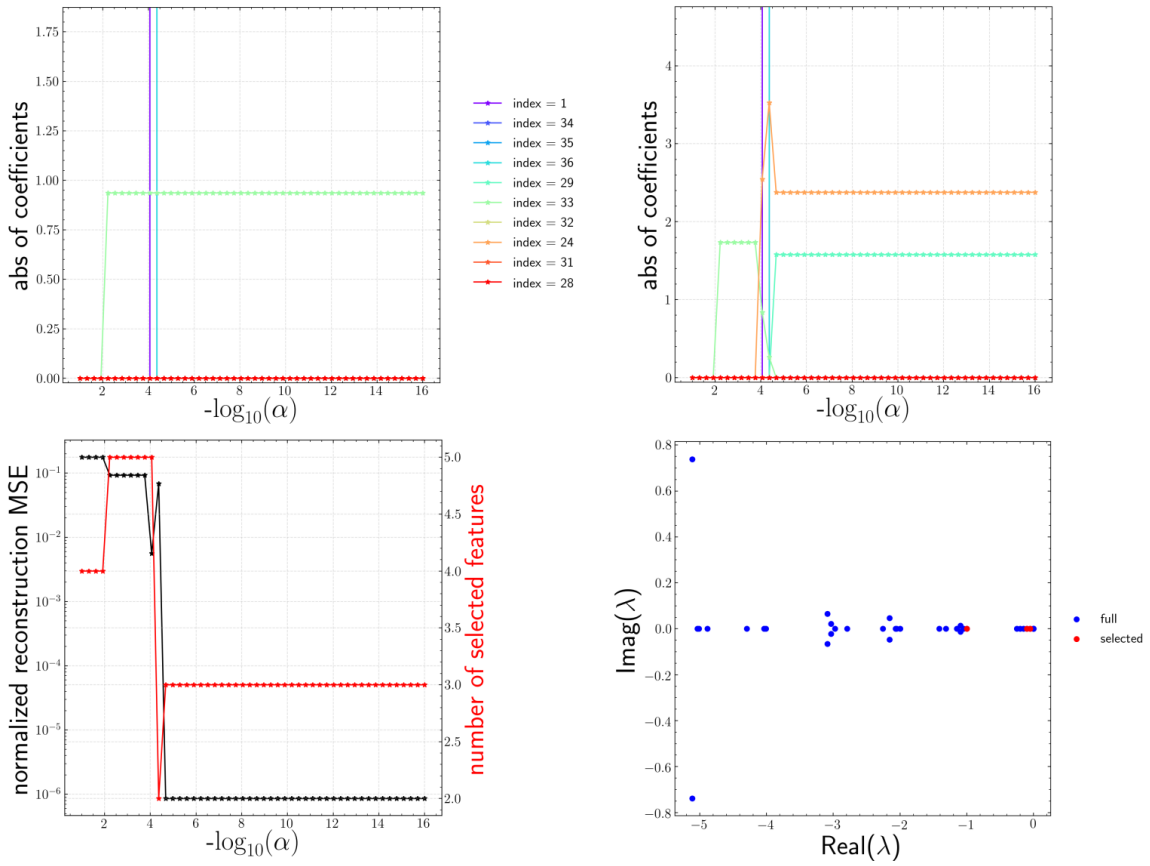


Figure 4.7:
Result of multi-task feature learning on top $\hat{L} = 10$ accurate eigenmodes from continuous-time EDMD for the 2D fixed point attractor. Top left: ElasticNet path for x_1. Top right: ElasticNet path for x_2. Bottom left: trends of normalized reconstruction error and number of non-zero terms versus α. Bottom right: selected continuous-time eigenvalues.

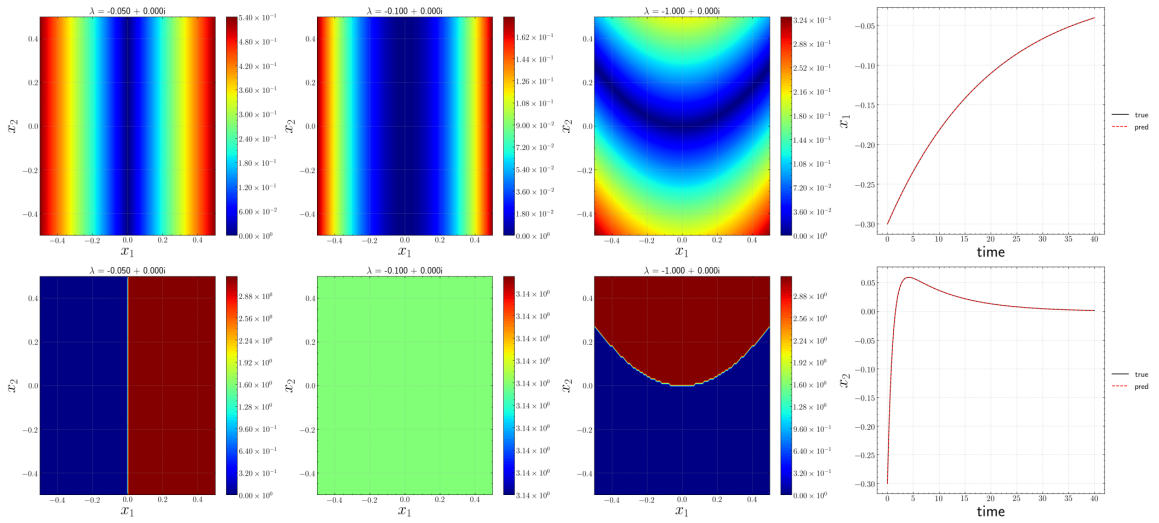


Figure 4.8: Sparsely selected eigenfunctions and eigenvalues from continuous-time EDMD for 2D fixed point attractor with corresponding prediction on testing data with an unseen initial condition $x_1(0) = x_2(0) = -0.3$. From left to right, the top three figures show contours of magnitude of eigenfunctions, while the bottom three figures are those of phase angle of eigenfunctions. Last column: comparison between prediction and ground truth for an unseen testing trajectory.

testing data is also excellent. Note that the indices of true eigenfunctions φ_1, φ_2 and φ_3 ordered by Kou's criterion in eq. (4.15) are 8, 5 and 6. In this case, all of the true eigenfunctions are missing in the top 3 modes chosen by Kou's criterion. Indeed, the top 3 modes chosen by Kou's criterion have nearly zero eigenvalues.

4.4.1.2 Results of continuous-time KDMD with mode selection

The mode selection algorithm presented above can be applied in precisely the same form to KDMD, given a set of eigenfunctions and eigenvalues. Error analysis of eigenfunctions is shown in fig. 4.9, from which we choose $\hat{L} = 10$ as well. As before, eigenvalues ordered by maximal deviation from linear evolution are shown in the legend in the right subfigure in fig. 4.9. Again, in the left subfigure in fig. 4.9, we observe a sharp decrease in the reconstruction error after the 4 most accurate modes are included. This is expected, as the second to fourth most accurate modes

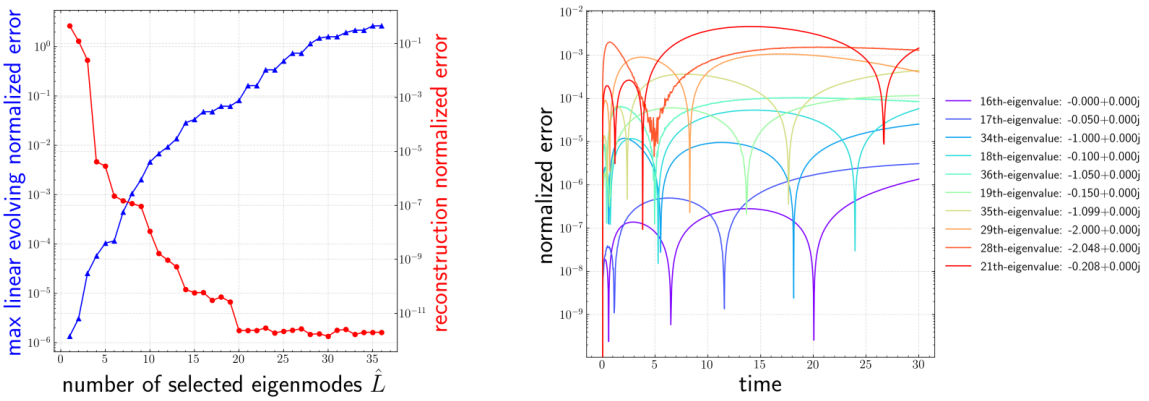


Figure 4.9: Error analysis of 36 eigenmodes from continuous-time KDMD for the 2D fixed point attractor. Left: trends of linearly evolving error Q and reconstruction error R. Right: temporal evolution of relative error for top $\hat{L} = 10$ accurate eigenmodes.

are analytically exact from the right subfigure. As shown in figs. 4.10 and 4.11, it is confirmed that that both spEDMD and spKDMD arrive at the same analytic eigenfunctions with difference up to a constant factor. It should be noted that, although polynomials are not analytically in the RKHS (*Minh*, 2010), good approximations can still be achieved *conditioned on the data* we have, i.e., $x_1, x_2 \in [-0.5, 0.5]$. Again, the indices of true eigenfunctions φ_1 to φ_3 ordered by Kou's criterion are 8, 2 and 3. Hence, φ_1 is missing in the top 3 modes chosen by Kou's criterion. Similarly, the first mode chosen by Kou's criterion has a zero eigenvalue.

4.4.1.3 Effect of SVD regularization

SVD truncation is a standard regularization technique in the solution of a potentially ill-conditioned linear system. In the standard EDMD in eq. (2.7) - for example - $\mathbf{G}$ could be potentially ill-conditioned, leading to spurious eigenvalues in $\mathbf{K}$. Hence, *Williams et al.* (2014) recommend SVD truncation in eq. (2.8) to obtain a robust solution of $\mathbf{K}$. Effectively, it shrinks the number of EDMD/KDMD modes. It has to be recognized, however, that the mode reduction from SVD truncation is not the same as mode selection. Most importantly, one should not confuse *numerical* spuriousness

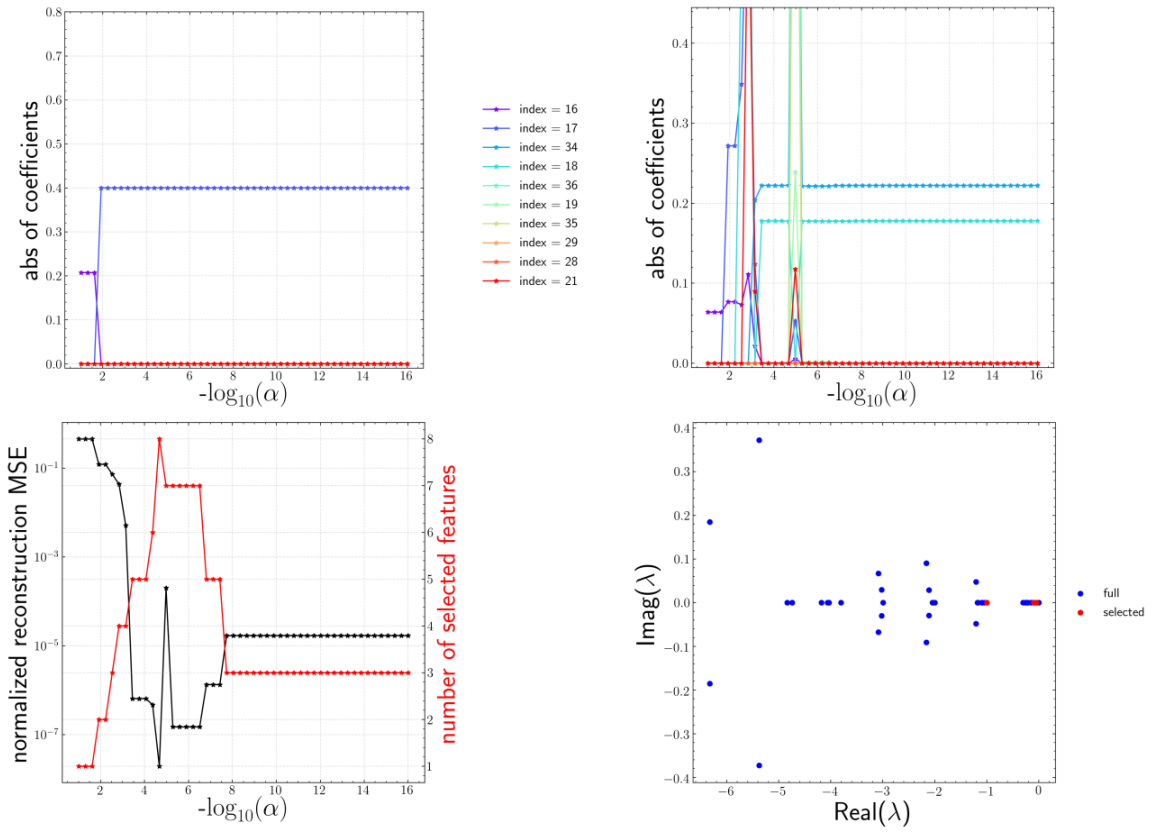


Figure 4.10:
Result of multi-task feature learning on top $\hat{L} = 10$ accurate eigenmodes from continuous-time KDMD for the 2D fixed point attractor. Top left: ElasticNet path for x_1. Top right: ElasticNet path for x_2. Bottom left: trends of normalized reconstruction error and number of non-zero terms versus α. Bottom right: selected continuous-time eigenvalues.

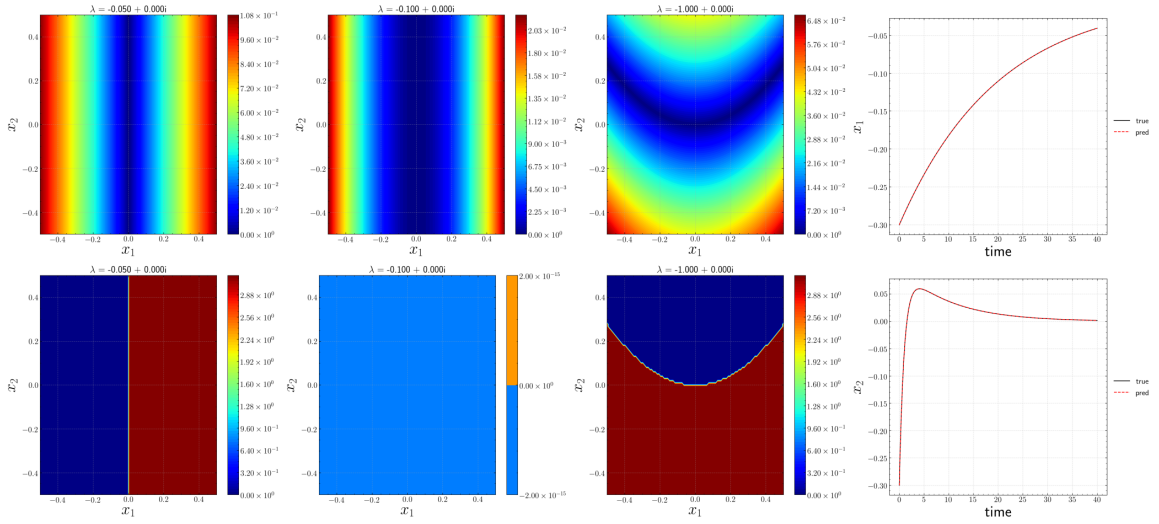


Figure 4.11: Sparsely selected eignefunctions and eigenvalues from continuous-time KDMD for 2D fixed point attractor with corresponding prediction on testing data with an unseen initial condition $x_1(0) = x_2(0) = -0.3$. From left to right, the top three figures show contours of the magnitude of eigenfunctions, while the bottom three figures are those of phase angle of eigenfunctions. Last column: comparison between predictions and ground truth for an unseen testing trajectory.

from poor numerical conditioning with *functional* spuriousness from the orthogonal projection error of the Koopman operator (*Korda and Mezić*, 2018a). Indeed, SVD truncation does not always lead to better approximation of a Koopman-invariant subspace. It is rather a linear dimension reduction that optimally preserves the variance in the feature space conditioned on the training data without knowing the linear evolution property of each feature.

For demonstration, we take the above fixed point attractor system where we use the same data and standard EDMD algorithm with the same order of Hermite polynomials. The results of prediction on the unseen testing data shown in fig. 4.12 indicate that even though only 3 eigenfunctions (indeed 3 feature in the Hermite polynomial) are required, standard EDMD fails to identify the correct eigenfunctions with 26 SVD modes while the results improve with 31 modes retained. The sensitivity of standard EDMD with respect to SVD truncation is likely a result of the use of normalized

Hermite polynomials where SVD truncation would lead to a strong preference over the subspace spanned by the higher order polynomials. We did not observe such a sensitivity for KDMD, unless the subspace is truncated below 10 modes.

4.4.2 Transient flow past a cylinder

As a classical example for Koopman analysis in fluid dynamics (*Bagheri*, 2013; *Williams et al.*, 2014; *Otto and Rowley*, 2019b), transient two-dimensional flow past cylinder (fig. 4.13) is considered at different Reynolds numbers ($Re = U_\infty D/\nu$), where $U_\infty = 1$ is the freestream velocity, $D = 2$ is the diameter of the cylinder, and ν is the kinematic viscosity. The two-dimensional incompressible Navier–Stokes equations govern the dynamics with far-field boundary conditions for pressure and velocity and no-slip velocity on the cylinder surface. Numerical simulations are performed using the `icoFoam` solver in `OpenFOAM` (*Jasak et al.*, 2007) solving the 2D incompressible Navier-Stokes equations. We explore $Re = 70, 100, 130$ by changing the viscosity. The pressure field is initialized with i.i.d Gaussian noise $\mathcal{N}(0, 0.3^2)$. The velocity is initialized with a uniform freestream velocity superimposed with i.i.d Gaussian noise $\mathcal{N}(0, 0.3^2)$. It should be noted that the noise is generated on the coarsest mesh shown in fig. 4.13, and interpolated to the finer meshes. Grid convergence with increasing mesh resolution is assessed by comparing the temporal evolution of the drag coefficient C_D and lift coefficient C_L.

Note that the dynamics of a cylinder wake involves four regimes: near-equilibrium linear dynamics, non-linear algebraic interaction between equilibrium and the limit cycle, exponential relaxation rate to the limit cycle, and periodic limit cycle dynamics (*Chen et al.*, 2012; *Bagheri*, 2013). Instead of considering data only from each of these regimes separately (*Chen et al.*, 2012; *Taira et al.*, 2019) or with only the last two regimes where exponential linear dynamics is expected (*Bagheri*, 2013), we start collecting data immediately after the flowfield becomes unstable, and stop after

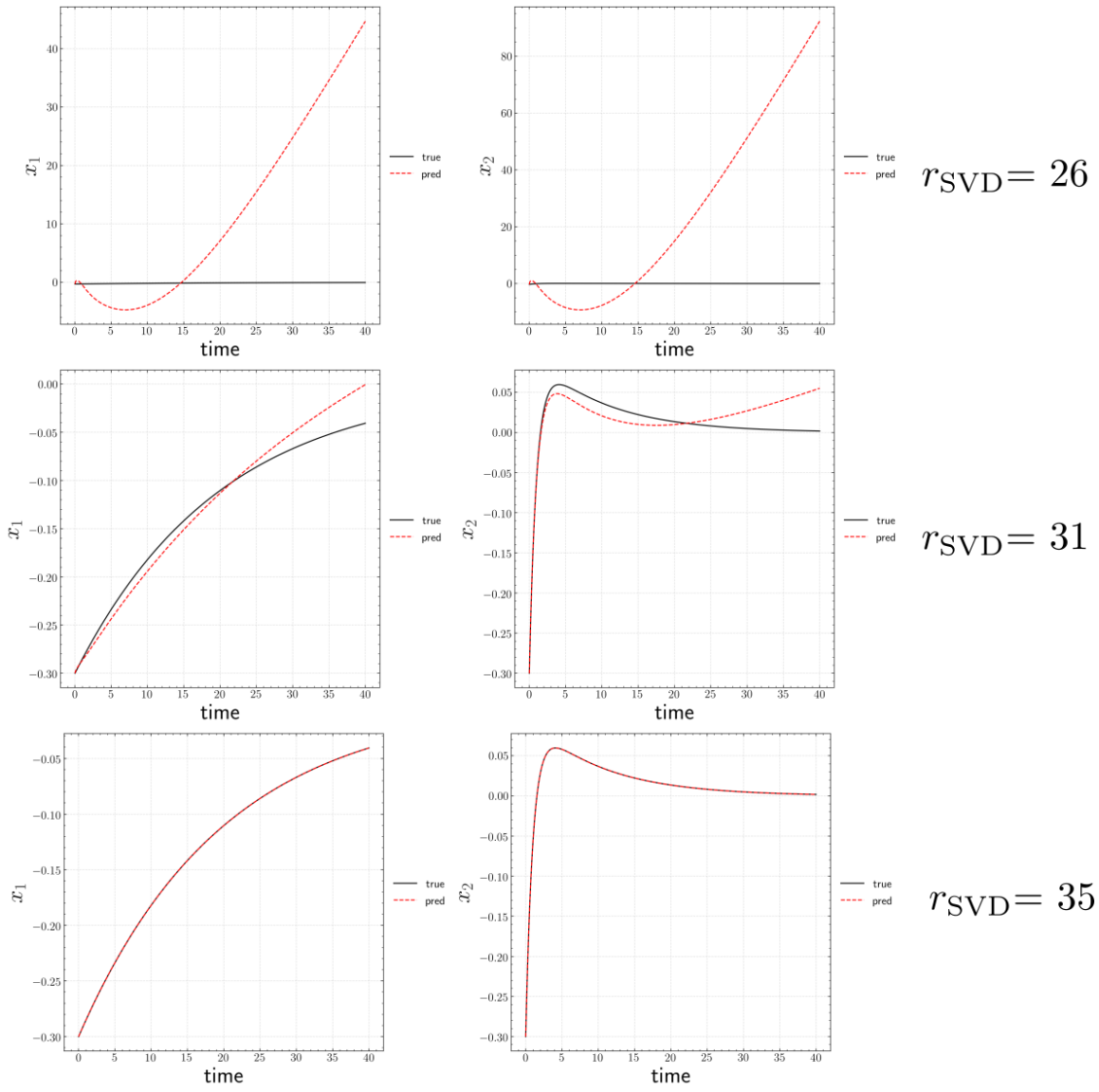


Figure 4.12: Standard EDMD prediction on unseen trajectory with different SVD truncations for fixed point attractor.

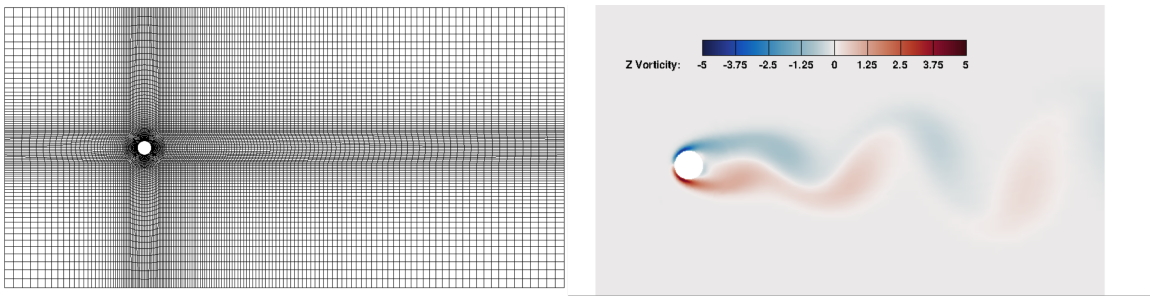


Figure 4.13: Left: illustration of computational mesh for 2D cylinder wake problem (coarsest). Right: contour of vorticity ω_z for $Re = 70$ when vortex shedding is fully developed (t = 175).

the flowfield experiences several limit cycles. Note that the regime with algebraic interaction is non-modal (*Schmid*, 2007), and therefore cannot be expressed as individual exponential terms (*Bagheri*, 2013). This becomes a challenging problem for DMD (*Chen et al.*, 2012). The sampling time interval is $\Delta t = 0.1 t_{ref}$ where $t_{ref} = D/U_\infty$.

For each Re, 891 snapshots of full velocity field U and V with sampling time interval Δt are collected as two matrices of size $N_{\text{grid}} \times N_{\text{snapshots}}$. Following this, each velocity component is shifted and scaled (normalized) between $[-1, 1]$. Since space-filling sampling in any high-dimensional space would be extremely difficult, we split the trajectory into training, validation, and testing data by sampling with strides similar to the "even-odd" sampling scheme previously proposed by *Otto and Rowley* (2019b). As illustrated in fig. 4.14, given index i, if $i \mod 3 = 0$, the i-th point belongs to training set while $i \mod 3 = 1$ corresponds to validation, and $i \mod 3 = 2$ for testing data. Consequently, the time interval in the training, testing, validation trajectory is tripled as $3\Delta t = 0.3 t_{ref}$. Thus, training, validation, and testing data are split into 297 snapshots each. Finally, we stack data matrices along the first axis corresponding to the number of grid points, and perform a distributed SVD described in section 4.3.5. For all three cases, the top 20 POD modes are retained, corresponding to 99% of kinetic energy. Next, we apply our algorithm to

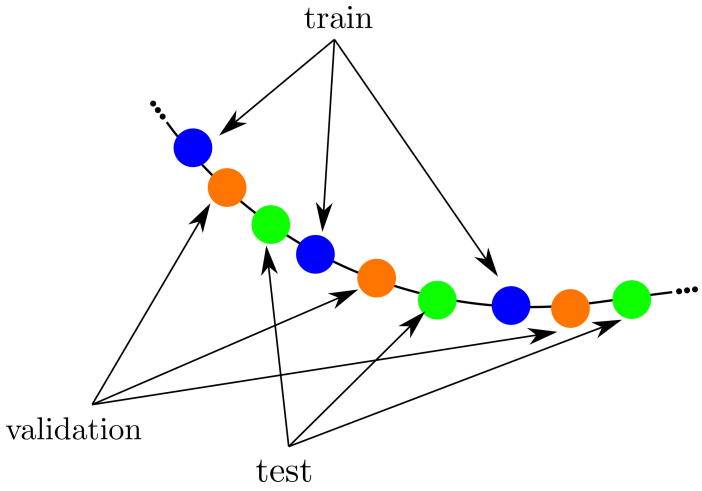


Figure 4.14: Illustration of splitting a uniformly sampled single trajectory in high dimensional phase space into training, validation and testing sets.

discrete-time KDMD with isotropic Gaussian kernel on this reduced-order non-linear system. We choose the hyperparameters $\sigma = 3$ and $r = 180$. Further details are given in appendix D.2.

4.4.2.1 Results of discrete-time KDMD with mode selection

For all three Reynolds numbers, a posteriori error analysis is shown in fig. 4.15. A good choice of the number of accurate modes $\hat{L}$ retained for reconstruction is around 60 since the corresponding maximal deviation from linear evolution is still around 5% while the reconstruction error reaches a plateau after $\hat{L} > 60$.

After the mode selection on validation data, a α-family of solutions is obtained with corresponding reconstruction error and the number of non-zeros terms as shown in fig. 4.16. Note that the chosen solution is highlighted as blue circles. As shown in table 4.1, nearly half of the accurate KDMD eigenmodes identified are removed with the proposed sparse feature selection. Note that for all three cases, the number of selected modes (around 32 to 34) is still larger than that required in neural network models (around 10) (*Pan and Duraisamy*, 2020a; *Otto and Rowley*, 2019b). This is because the subspace spanned by KDMD/EDMD relies on a pre-determined dictio-

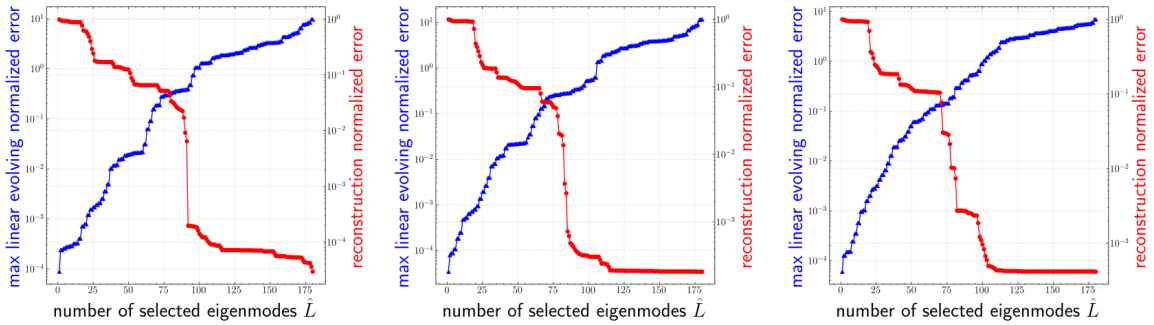


Figure 4.15: Trend of linear evolution error Q and reconstruction error R from discrete-time KDMD for the 20D cylinder wake flow case. Left: $Re = 70$. Center: $Re = 100$. Right: $Re = 130$.

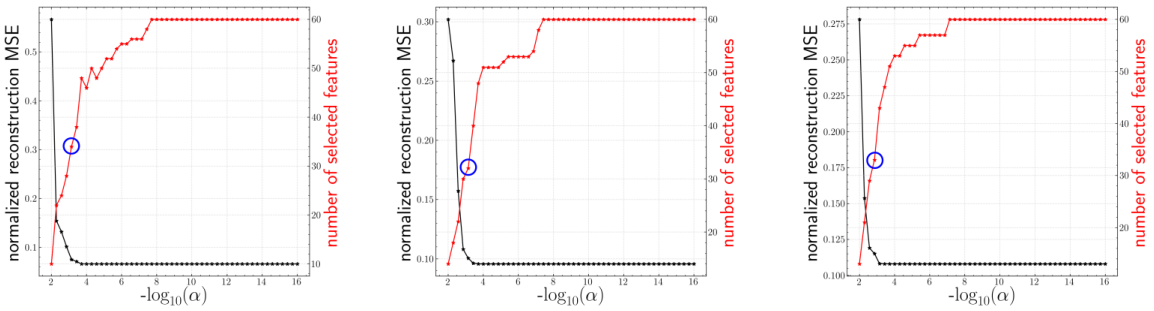


Figure 4.16: Variation of reconstruction error R and number of non-zero terms for the 20D cylinder wake flow. Left: $Re = 70$. Center: $Re = 100$. Right: $Re = 130$. Blue circle corresponds to selected α.

nary rather than being data-adaptive like neural network models. Nevertheless, due to the additional expressiveness from non-linearity, we will see in section 4.4.2.3 that spKDMD performs significantly better than DMD (*Schmid*, 2010) and spDMD (*Jovanović et al.*, 2014), while enjoying the property of convex optimization at a much lower computational cost than the inherently non-convex and computationally intensive neural network counterparts.

The predictions of the top 8 POD coefficients (denoted as x_1 to x_8) on testing data are displayed in figs. 4.17 to 4.19. The results match very well with ground truth for all three cases. Figure 4.21 shows that there appear to be five clusters of selected eigenvalues while most of the modes are removed by the proposed algorithm.

Table 4.1: Summary of mode selection on 20D cylinder wake flow.

	$Re = 70$	$Re = 100$	$Re = 130$
α_{select}	7.19×10^{-4}	7.19×10^{-4}	1.39×10^{-3}
number of selected modes	34	32	33
number of total modes	297	297	297
normalized reconstruction error R	0.075	0.105	0.113

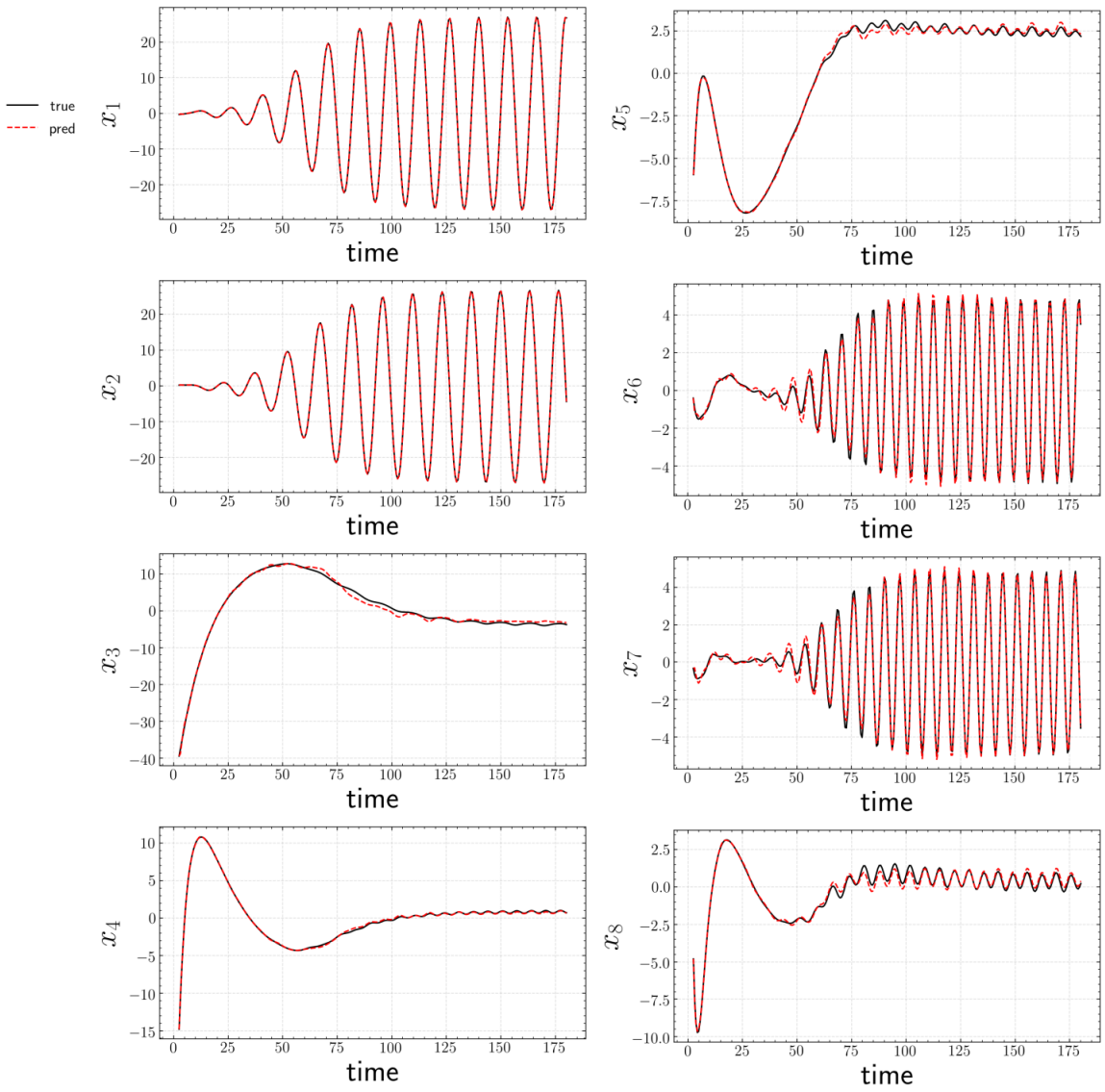


Figure 4.17: A posteriori prediction of testing trajectory for $Re = 70$ in terms of top 8 POD coefficients with sparsity-promoting KDMD.

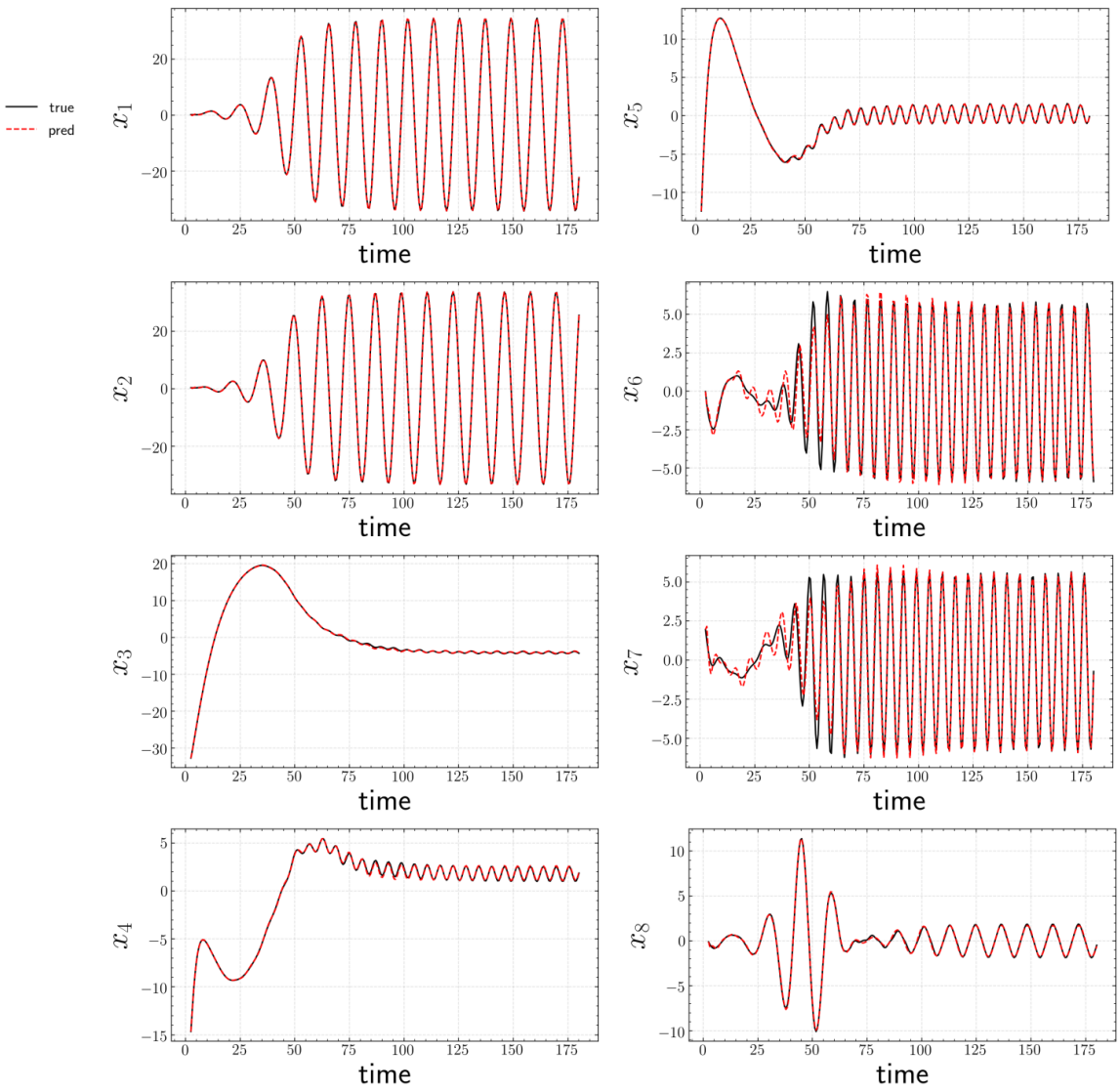


Figure 4.18: A posteriori prediction of testing trajectory for $Re = 100$ in terms of top 8 POD coefficients with sparsity-promoting KDMD.

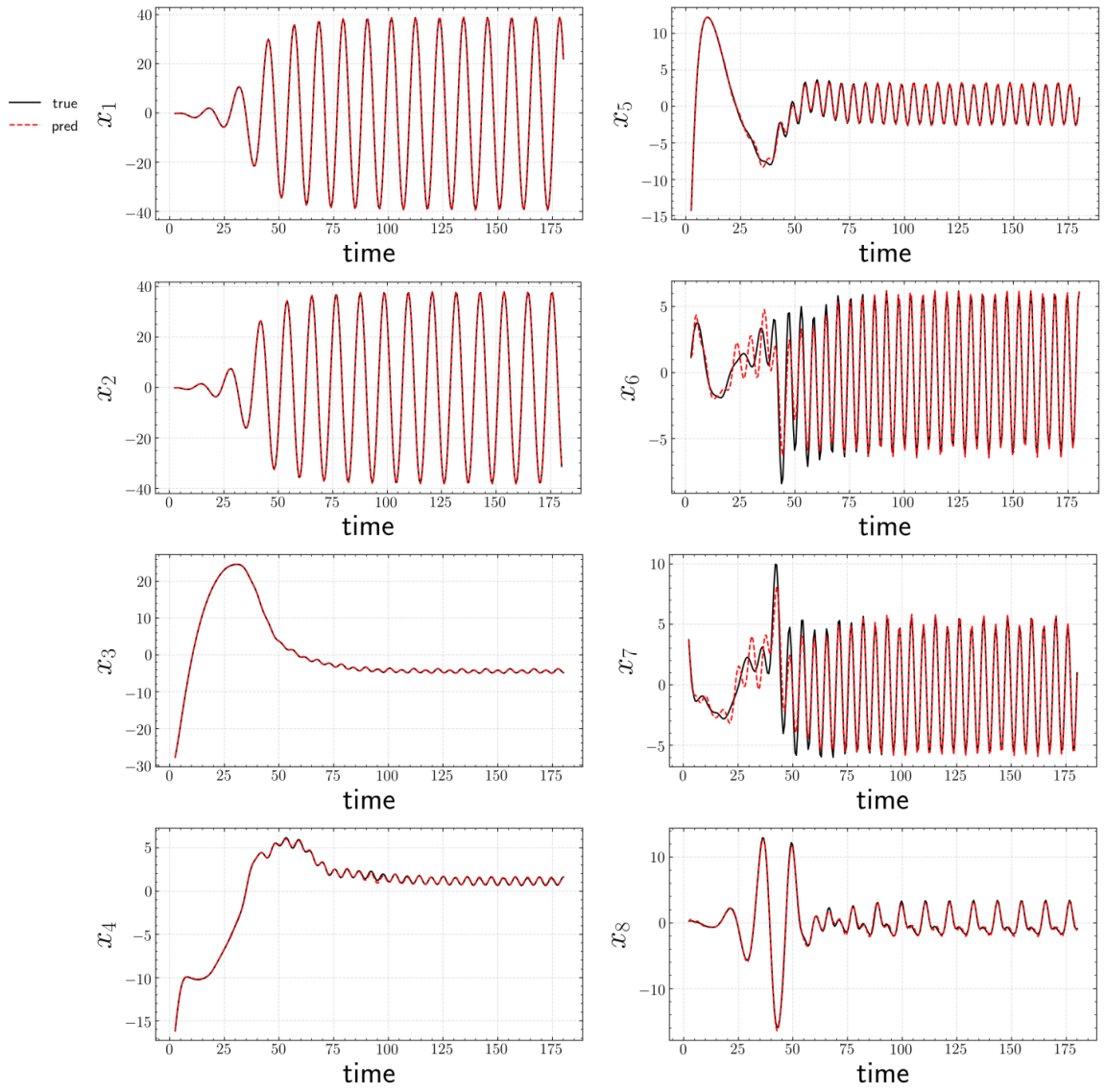


Figure 4.19: A posteriori prediction of testing trajectory for $Re = 130$ in terms of top 8 POD coefficients with sparsity-promoting KDMD.

Similar observations were also made in (*Bagheri*, 2013) when DMD is applied to the *full* transient dynamics. This pattern consisting of a stable eigenvalue *on* the unit circle surrounded by several decaying eigenvalues is observed for all clusters. The stable eigenvalue contributes to limit-cycle behavior, while the decaying eigenvalues account for the transient phenomenon. Due to symmetry, only eigenvalues in the first quadrant are shown in the bottom row of fig. 4.21. It is observed that the frequency associated with the type-II cluster is approximately twice that of type-I. This is in good agreement with previous analytical results from the weakly non-linear theory (*Bagheri*, 2013). The frequency f is normalized as $\mathrm{St} = fD/U_\infty$, where St is the Strouhal number.

Recall that in the laminar parallel shedding region ($47 < Re < 180$), the characteristic Strouhal number St scales with $-1/\sqrt{Re}$ (*Fey et al.*, 1998). Therefore, it is expected that St of both types tend toward higher frequency as Re increases from 70 to 130. Further, it is interesting to note that the corresponding Strouhal numbers for lift and drag when the system is on the limit-cycle - St_L and St_D[3] - coincide with the stable frequency of type-I and II respectively as indicated in fig. 4.21. This is due to the anti-symmetrical/symmetrical structure of the velocity field of type-I/II Koopman mode respectively as can be inferred from fig. 4.22. A schematic is is also shown in fig. 4.20.

The higher frequency mode is symmetrical (along the freestream direction) in U and anti-symmetrical in V. As a consequence, this only contributes to the oscillation of drag. The lower frequency mode is anti-symmetrical in U and symmetrical in V, and only contributes to the oscillation of lift. Thus, the fluctuation in the lift mostly results from the stable mode in type-I, while that for drag results from the stable mode in type-II with twice the frequency.

Finally, several representative Koopman modes from spKDMD for three Re are

[3]we observe that each of lift and drag coefficients exhibits only one frequency at limit-cycle regime for the range of Re studied in this work.

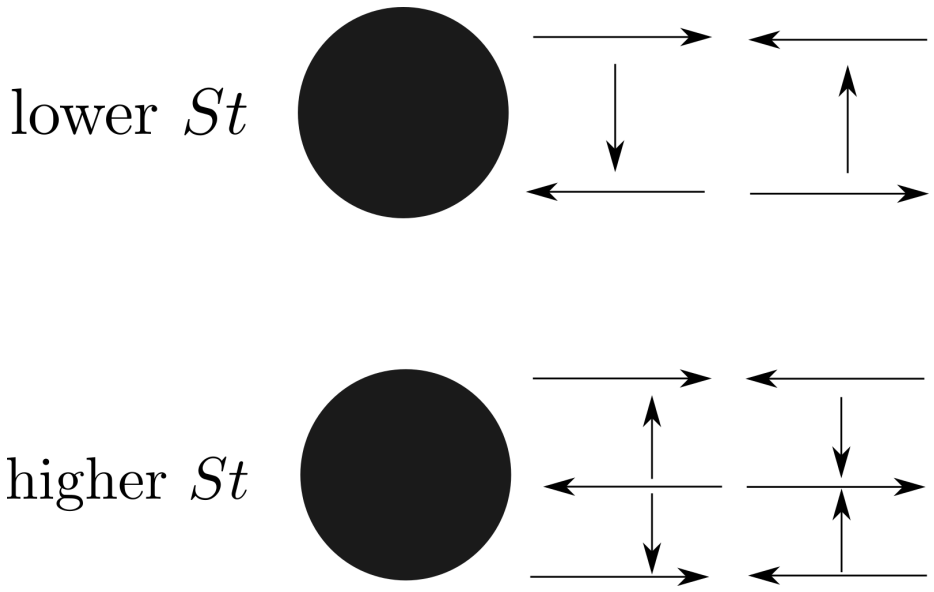


Figure 4.20: Illustration of the structure of velocity field for the lower (top) and higher frequency (bottom) Koopman modes. The arrow roughly indicates the velocity direction.

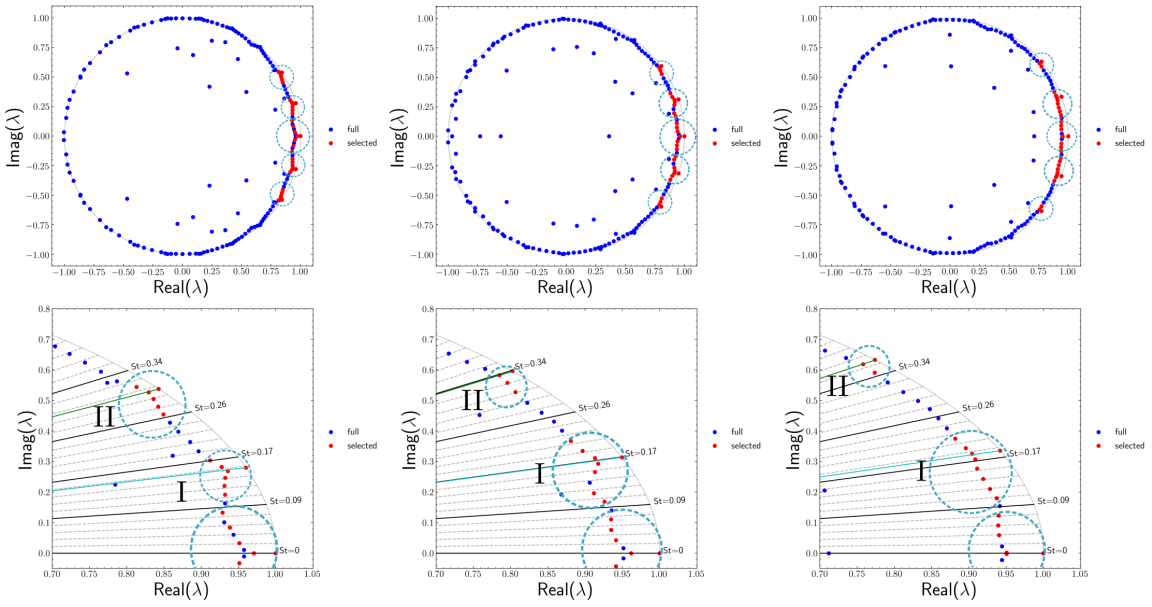


Figure 4.21: Discrete-time eigenvalue distribution of full KDMD and spKDMD. Left: $Re = 70$. Center: $Re = 100$. Right: $Re = 130$. Blue dot: full KDMD eigenvalues. Red dot: spKDMD eigenvalues. Bottom row: zoomed. I and II correspond to two types of eigenvalue clusters of distinct frequencies, with each of them enclosed by cyan dashed circles. Green/cyan solid line correspond to St_D/St_L.

shown in figs. 4.22 to 4.24. For a better comparison of mode shapes, contributions from the stable modes of type-I and II with a threshold of 0.001 at $t = 0$ is displayed in the top left of fig. 4.25. To remove the effect of time, the "envelope" of the mode shape, i.e., time average of the iso-contours is shown in the top right of fig. 4.25. From these results, we observe the following interesting phenomena:

- the minimal dimension of the Koopman-invariant subspace that approximately captures the limit cycle attractor for all three Re that fall into laminar vortex shedding regime (*White and Corfield*, 2006) is five, which is consistent with previous multi-scale expansion analysis near the critical Re (*Bagheri*, 2013).

- the lobes of stable Koopman modes in the type-I cluster show an approximately 50% larger width than those in a type-II cluster.

- similarity/colinearity among Koopman modes within each cluster is observed. Such a finding is previously reported in the theoretical analysis by *Bagheri* (2013). A similarity in spatial structure exists among the Koopman modes belonging to the same family, even though the structures are clearly phase lagged.

- as Re increases from 70 to 130, mode shapes flatten downstream while expand upstream.

- at $Re = 70$, the shear layer in the stable Koopman modes continues to grow within the computational domain. However, at $Re = 100, 130$, the shear layer stops growing after a certain distance that is negatively correlated with Re.

4.4.2.2 Net contribution of clustered Koopman modes

By examining figs. 4.22 to 4.24, we observe that the colinearity of the spatial structures among each cluster can cause cancellations. A famous example of such

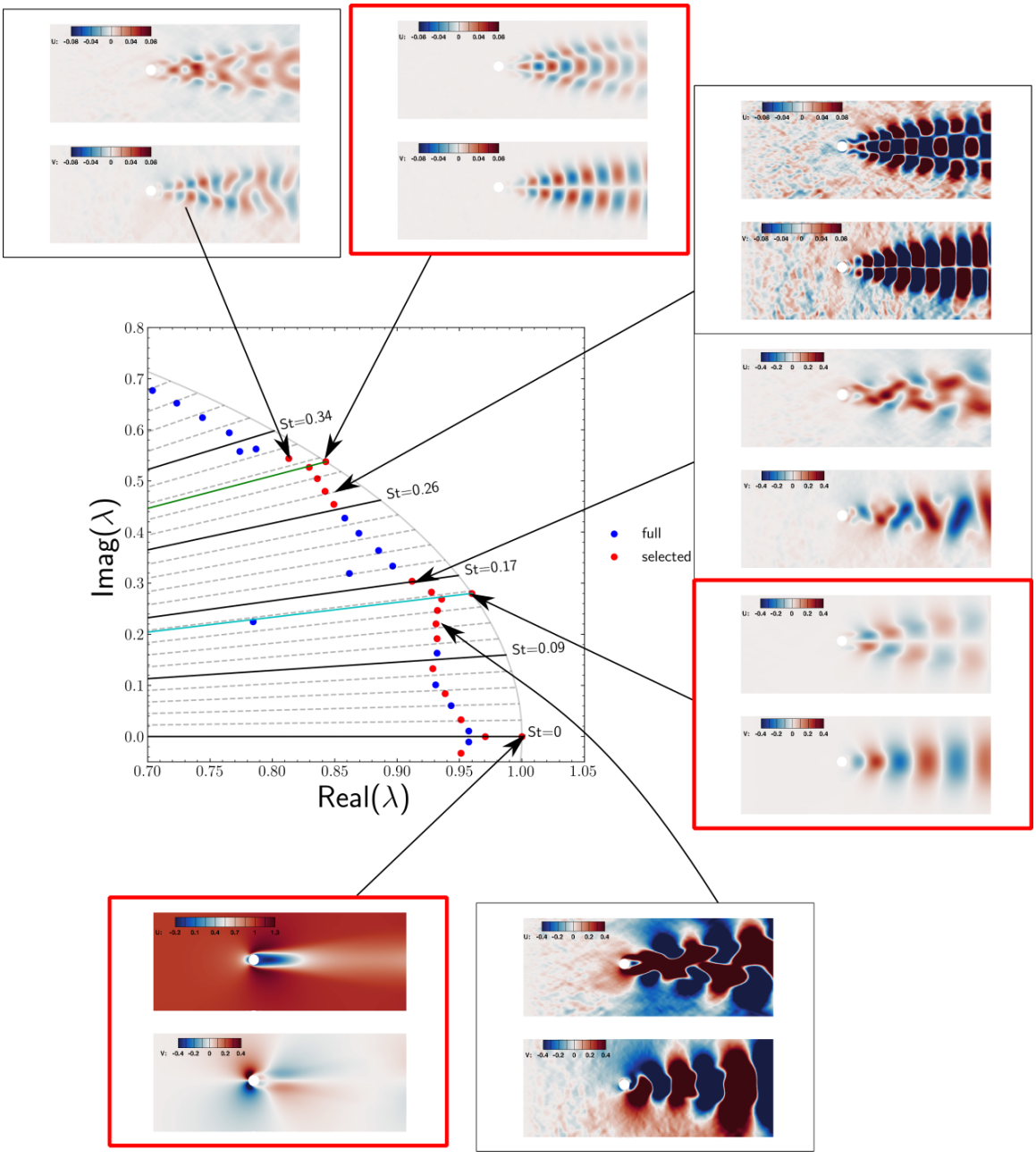


Figure 4.22: Contours of Koopman modes of $Re = 70$ cylinder wake flow at $t = 0$. Red squares indicate stable modes.

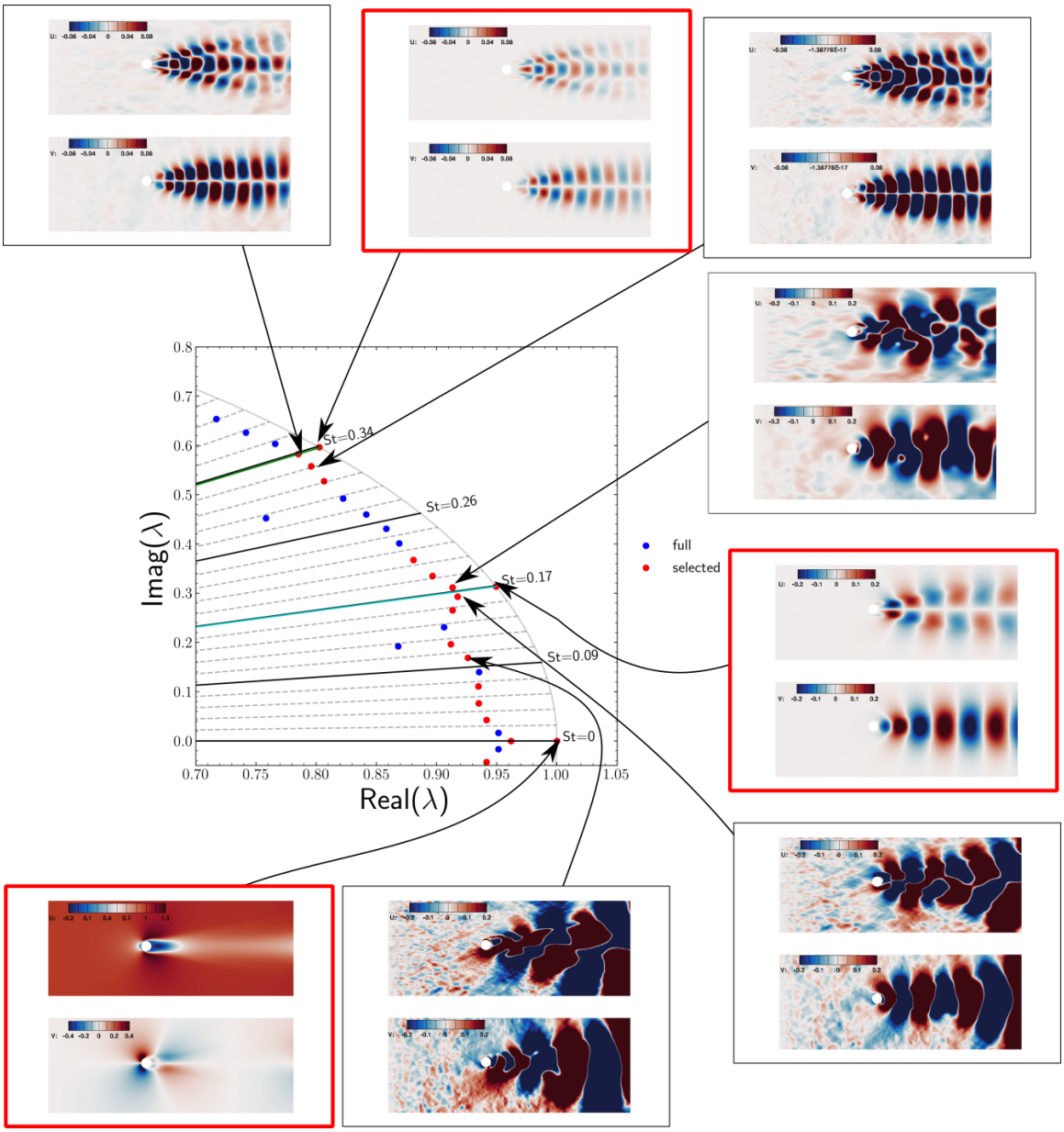


Figure 4.23: Contours of Koopman modes of $Re = 100$ cylinder wake flow at $t = 0$. Red squares indicate stable modes.

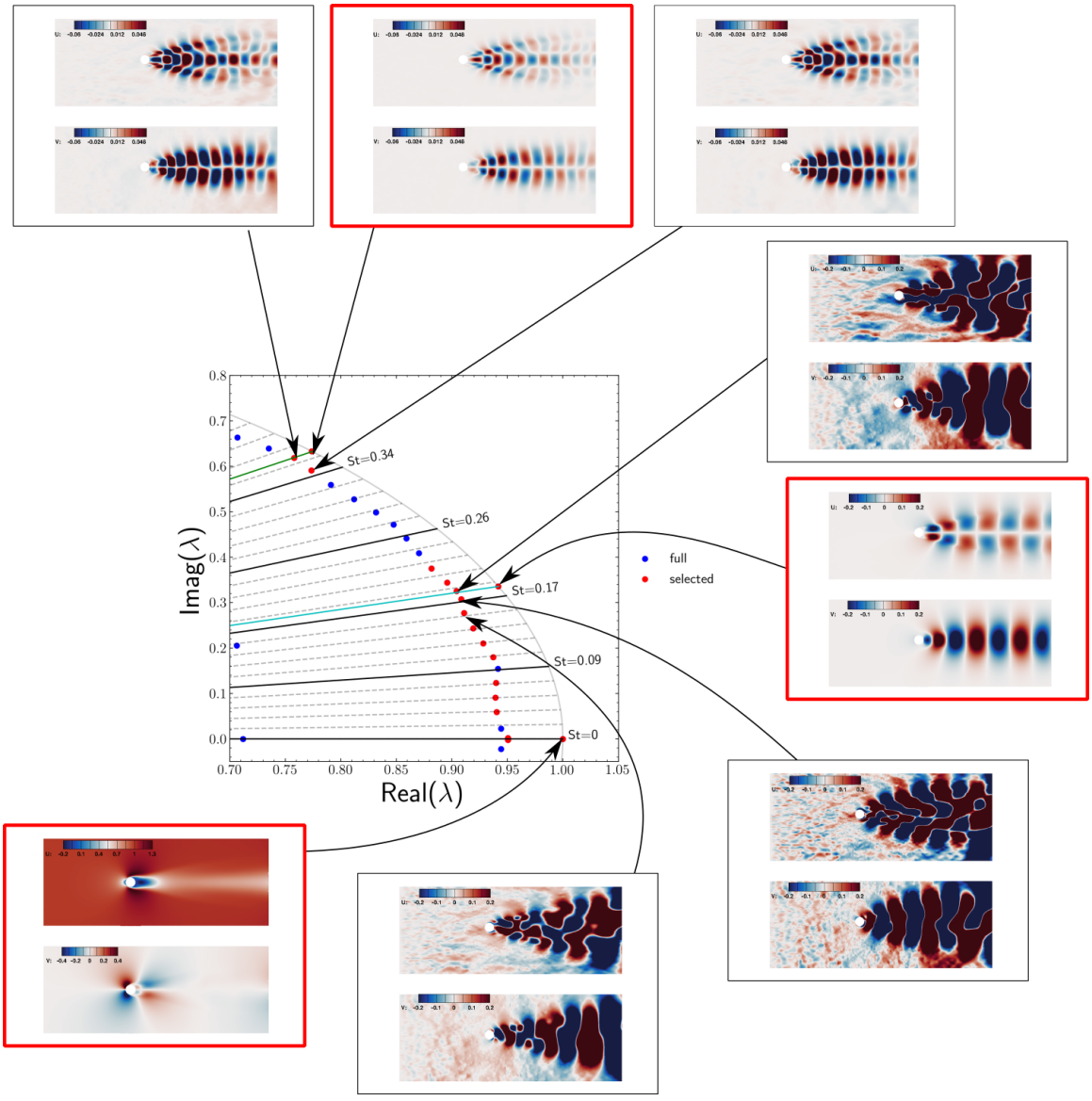


Figure 4.24: Contours of Koopman modes of $Re = 130$ cylinder wake flow at $t = 0$. Red squares indicate stable modes.

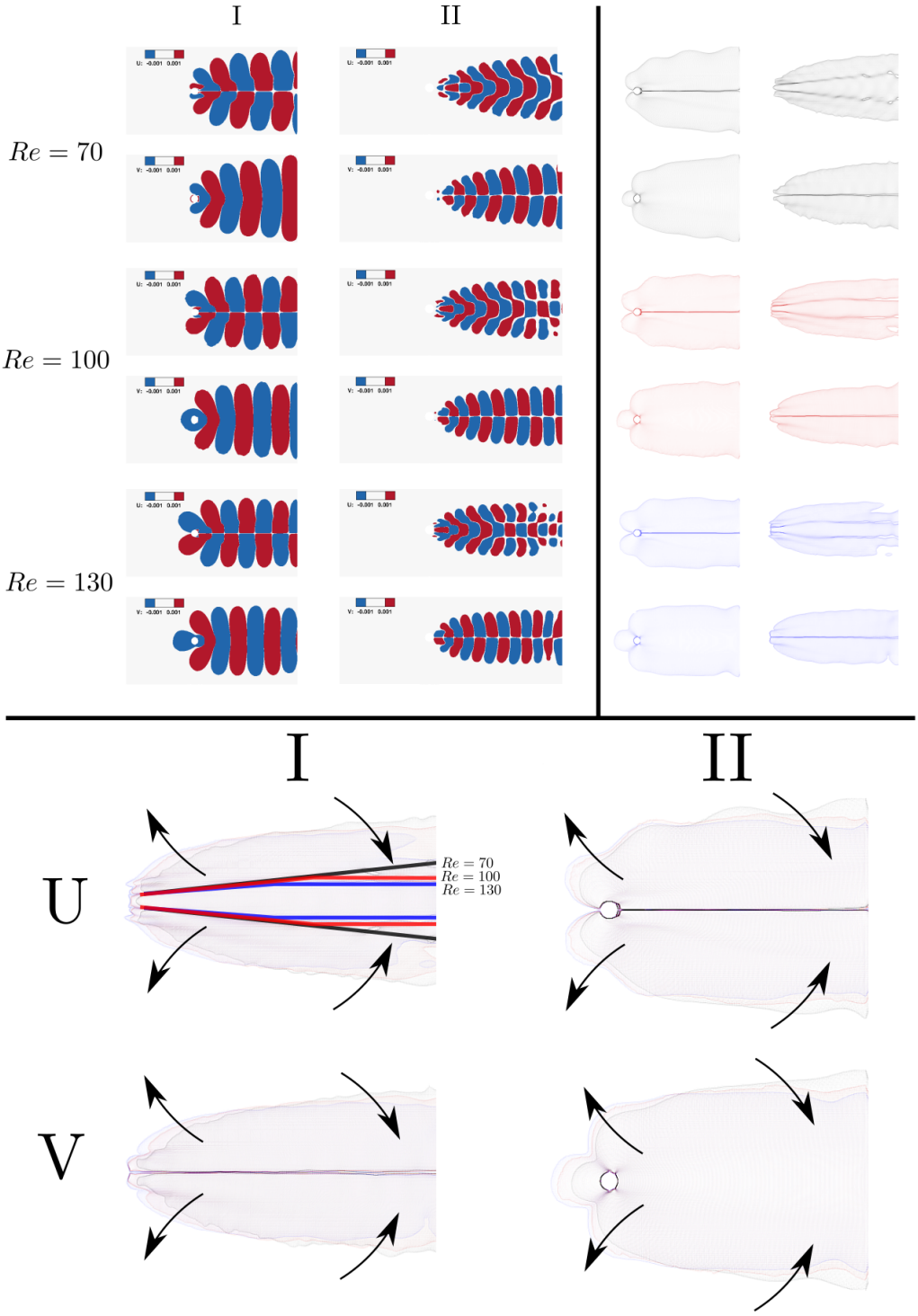


Figure 4.25: Top left: contribution of stable Koopman modes corresponding to type-I and type-II cluster for $Re = 70, 100, 130$ at $t = 0$ visualized with threshold 0.001. Top right: time-averaged iso-contour of top left plot. Bottom: tendency of "envelope" of type-I and II modes as Re increases. Separation lines in U component of type-I are drawn for $Re = 70$ (black), $Re = 100$ (red) and $Re = 130$ (blue).

non-oscillatory cancellation is the "shift mode" defined by *Noack et al.* (2003), in conjunction with two oscillating modes. As the "shift mode" decays, the stationary component of the flow transitions from the unstable equilibrium to the time-averaged mean. The existence of such non-oscillatory decaying Koopman modes is also confirmed by weakly non-linear analysis (*Bagheri*, 2013). Interestingly, our algorithm is able to identify not only the non-oscillatory cancellation (from the "shift mode") but also oscillatory cancellations from two clusters with distinct frequencies. Such cancellations elegantly explain why no unstable Koopman eigenvalues appear in this flow given the co-existence of an attracting limit cycle and an unstable equilibrium. These modes could be understood as "oscillatory shift modes", as a generalization of the model proposed by *Noack et al.* (2003).

Since modes within each cluster appear to be colinear to each other, it is intriguing to investigate the net contribution from each cluster. For the above $Re = 70$ case, effects from different clusters for different time in the transient regime are shown in fig. 4.26. There are several interesting observations throughout the transient period from $t = 80$ to $t = 200$:

- The net contribution from "cluster 0" does not exhibit strong oscillations. For the contribution from "cluster 0 ", the U component shows a decrease in the length of the reverse flow region behind the cylinder with an increase in the low speed wake layer thickness while the V component remains unchanged. This is similar to the effect of "shift mode" which also characterizes the decay of recirculation behind the cylinder.

- Initially at $t = 80$, the net contribution from "cluster I" is rather weak primarily due to the "cancellation" from the lagged phase. Further, the development of vortex shedding downstream from the top and bottom surfaces of the cylinder is nearly parallel. This corresponds to the initial wiggling of the low speed

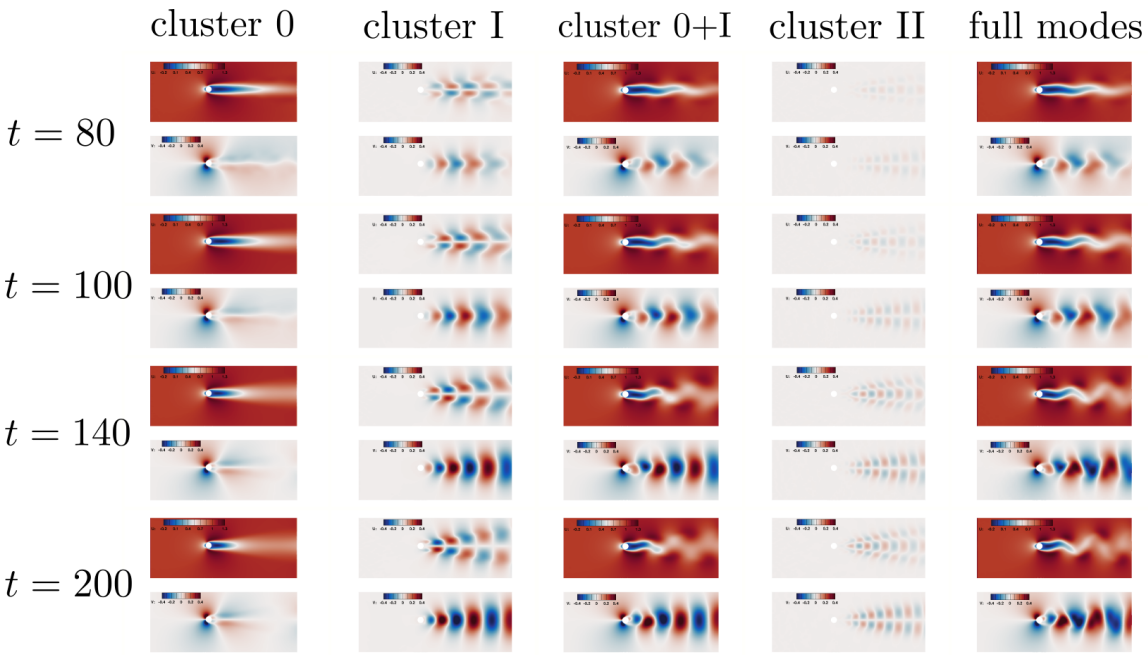


Figure 4.26: Contribution of Koopman modes at cluster level in the transient regime of $Re = 70$ case. "cluster 0" denotes the cluster near the real axis in fig. 4.21. "cluster I" /"cluster II" takes the effect of mirror cluster in fourth quadrant into account. "full modes" denotes the aggregated contribution of Koopman modes.

wake flow downstream. As time increases, the pattern of corresponding vortices develops away from the center line.

- Although the net contribution from "cluster 0+I" captures most of the flow features from "full modes" throughout the transient regime, with increasing time, the net contribution from "cluster II" becomes more important and contributes to the strength of vortex shedding downstream.

4.4.2.3 Comparison with DMD and sparsity-promoting DMD at $Re = 70$

To confirm the advantage of sparsity-promoting KDMD over DMD (*Schmid*, 2010) and spDMD (*Jovanović et al.*, 2014), we compare the following three models on the unsteady cylinder wake flow at $Re = 70$:

1. sparsity-promoting KDMD on top 20 POD coefficients with 34 modes selected,

2. DMD on top 20 POD coefficients,

3. spDMD [4] on top 200 POD coefficients with α chosen carefully such that only 34 modes are selected.

Note that DMD with top 200 POD coefficients, i.e., $r = 200$ in SVD-DMD (*Schmid*, 2010), contains 10 times stable/decaying harmonics as DMD on the top 20 modes. Hence, it is not surprising to expect that the corresponding prediction of the evolution of the top 20 POD coefficients to be very good (not shown for clarity). However, to make a fair comparison against sparsity-promoting KDMD, we consider spDMD (*Jovanović et al.*, 2014) on the top 200 POD coefficients with 34 modes selected[5].

As shown in fig. 4.27, given the same number of eigenmodes, sparsity-promoting KDMD performs remarkably better than spDMD, especially in the transient regime. This is likely due to the inability of DMD in capturing non-linear algebraic growth in the transient regime (*Bagheri*, 2013). spDMD overestimates the growth in x_1 and x_2 while ignoring a turnaround near the onset of transient regime in x_5 and x_8. As expected, DMD with 20 POD coefficients performs the worst especially for the modes where transient effects are dominant. Given the results in fig. 4.27, among all of the top 8 POD coefficients, x_6 and x_7 appear to be most challenging to model: DMD and spDMD cannot match the limit-cycle while spKDMD performs very well. Notice that the frequency in x_6 and x_7 corresponds to St_D. Hence, there will be a difference in predicting the fluctuation of C_D between spDMD and sparsity-promoting KDMD.

Finally, comparison of the identified Koopman eigenvalues between DMD, spDMD and spKDMD is shown in fig. 4.28. On one hand, both spDMD and spKDMD exactly capture the stable eigenmodes that correspond to St_D and St_L. This is expected since

[4] We used the original Matlab code from http://people.ece.umn.edu/users/mihailo/

[5] Although there are 200 POD coefficients used for spDMD and 20 for KDMD, it is not an unfair comparison given that the same number of spatial modes are selected. Further, these are reduced order models of the same the full order dynamical system.

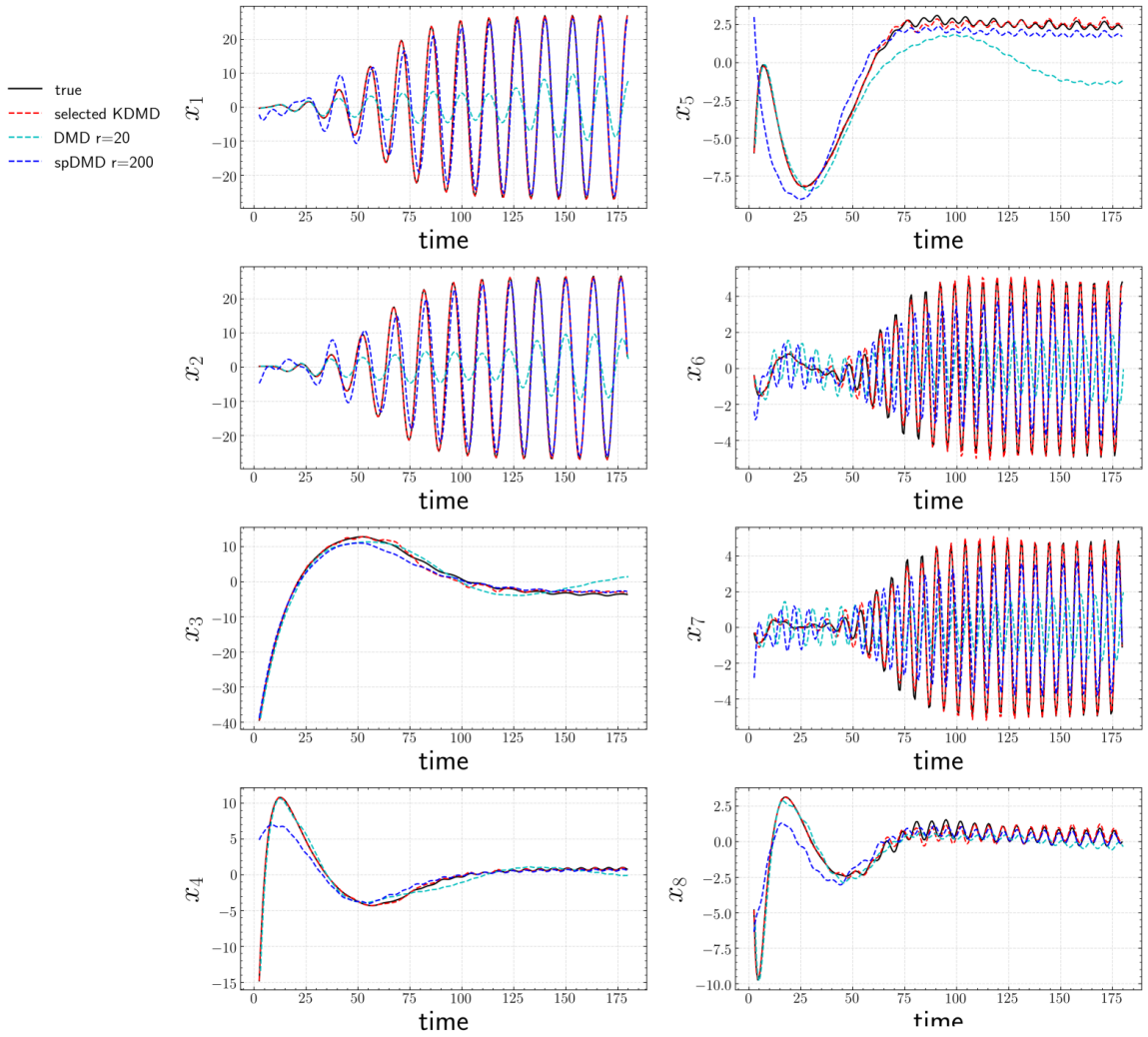


Figure 4.27: Comparison of a posteriori prediction on the top 8 POD coefficients of the testing trajectory between sparsity-promoting KDMD, DMD (*Schmid*, 2010) and spDMD (*Jovanović et al.*, 2014) for the 2D cylinder flow at $Re = 70$. x_i denotes i-th POD coefficient.

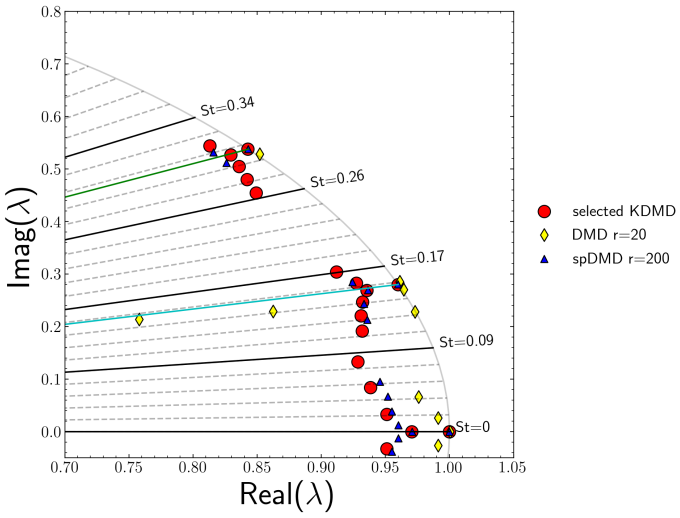


Figure 4.28: Comparison of identified eigenvalues between spKDMD, DMD (*Schmid*, 2010) and spDMD (*Jovanović et al.*, 2014) for the 2D cylinder flow at $Re = 70$

DMD with 200 POD coefficients represents the dynamics very well, and deviation from limit-cycle behavior would be penalized in spDMD. On the other hand, several erroneous stable DMD modes are obtained by DMD. This explains the deviation of a posteriori prediction from the ground truth limit cycle in fig. 4.27. For those decaying modes, similarity is observed between two clusters of eigenvalue from spDMD and spKDMD. However, spKDMD contains more high frequency modes than spDMD. Finally, it is interesting to note that, although the correct stable eigenvalues are captured accurately by both spDMD and spKDMD, the former does not capture accurate amplitudes for stable eigenvalues of type-II as seen in fig. 4.27.

As a side note, when the temporal mean was used instead of maximal error in the definition of Q in eq. (4.2), spKDMD with the above setting was found to not find a stable eigenvalue corresponding to ST_D.

4.4.3 Transient turbulent ship airwake

Understanding unsteady ship airwake flows is critical to design safe shipboard operations, such as takeoff and landing of fixed or rotary wing aircraft (*Forrest and*

Owen, 2010), especially when the wind direction becomes stochastic. Here we obtain snapshots from an unsteady Reynolds Averaged Navier–Stokes (URANS) simulation of a ship airwake using FLUENT (*Ansys*, 2016) with the shear layer corrected k-ω two-equation turbulence model. Unsteadiness arises from both bluff-body separation, and an abrupt change in wind direction. We consider a conceptual ship geometry called simple frigate shape 2 (SFS2). For the details of the geometry, readers are referred to (*Yuan et al.*, 2018). Configuration of the simulation setup is shown in fig. 4.29 where α_∞ denotes the angle of side wind.

To prepare a proper initial condition, a URANS simulation for $U_\infty = 15$m/s with $\alpha_\infty = 0^\circ$, i.e., no side wind, is conducted to reach a physical initial condition. Following this, the last snapshot is used as the initial condition for a new run with an impulsive change in the wind direction from $\alpha_\infty = 0^\circ$ to $\alpha_\infty = \alpha_0 = 5^\circ$. The boundary conditions for outlet/input is pressure outlet/velocity inlet while top and bottom are set as symmetry for simplicity. No-slip conditions are used at surface of the ship. Further details on the simulation setup are provided in (*Sharma et al.*, 2019).

The sampling time interval is $\Delta t = 0.1$s with 500 consecutive samples of the three velocity components. This corresponds to several flow through times over the ship length. The domain of interest is a cartesian region of mesh size $24 \times 40 \times 176$ starting on the rear landing deck. For dimension reduction, the trajectory of the top 40 POD coefficients (temporal mean subtracted) are collected, yielding $> 99\%$ kinetic energy preservation. Discrete-time KDMD with an isotropic Gaussian kernel is employed to perform non-linear Koopman analysis where $\sigma = 200$, $r = 135$ is chosen. Details of hyperparameter selection are provided in appendix D.3.

4.4.3.1 Results of discrete-time KDMD with mode selection

First, the error analysis of eigenfunctions is shown in fig. 4.30, where we choose $\hat{L} \approx 60$ for good reconstruction. However, the level of deviation from linear evolution

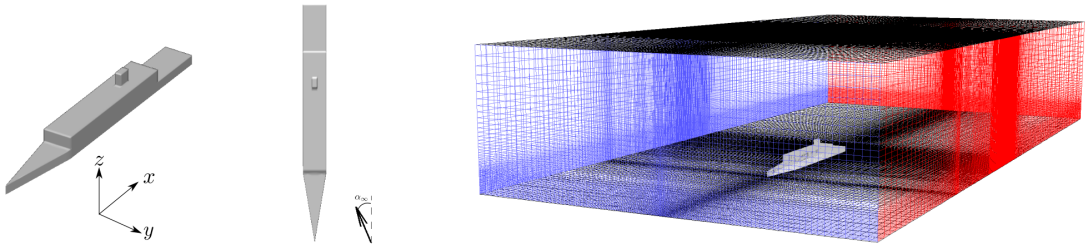


Figure 4.29: Left: geometry of the ship (SFS2). Right: generated computational mesh.

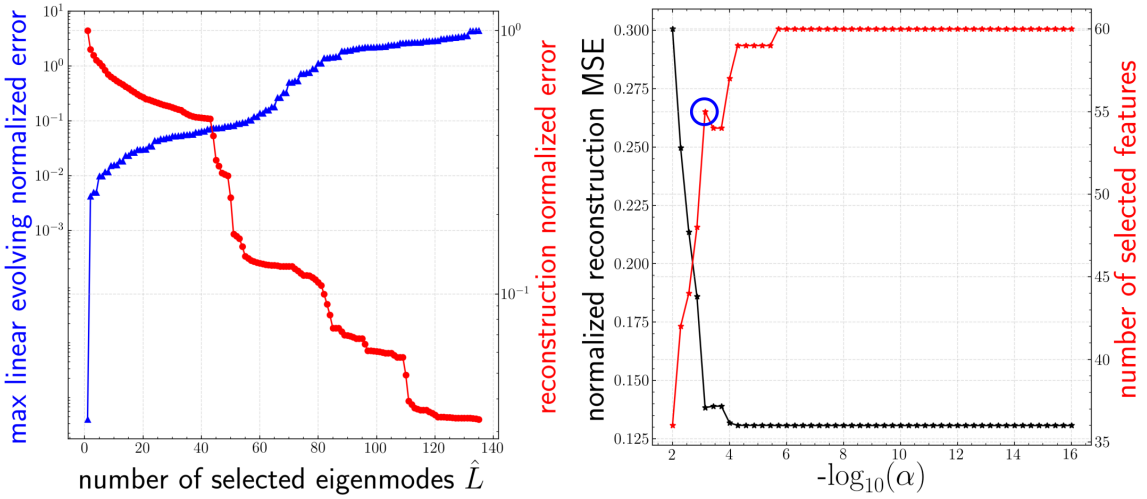


Figure 4.30: Left: Trend of linearly evolving error Q and reconstruction error R from discrete-time KDMD for the ship airwake. Right: Trend of linearly evolving error Q and reconstruction error R from discrete-time KDMD.

is around 10%. This error will be reflected later as deviation in a posteriori prediction on the testing trajectory[6].

Second, the result of mode selection is summarized in table 4.2. Note that nearly 2/3 modes are removed. Furthermore, model performance in terms of a posteriori prediction on the testing trajectory is evaluated. Comparison between KDMD and the ground truth on contours of velocity components on a special $z-$plane (1.2 meters above the landing deck) is shown in fig. 4.31. Effects of an impulse change in wind

[6]This implies difficulties in finding an accurate yet informative Koopman operator with isotropic Gaussian kernels. However, choosing an optimal kernel type is beyond the scope of this work.

Table 4.2: Summary of mode selection for ship airwake.

α_{select}	7.19×10^{-4}
number of selected modes	55
number of total modes	167
normalized reconstruction error R	0.133

direction in the following are observed from $t = 1.5s$ to $t = 30s$ and well-captured by spKDMD:

- growth of a large shear layer in U on the rear (left) side of the superstructure on the ship
- a strong side wind sweep in V above the landing deck propagating downstream
- development of vortex on the upwind (right) downstream side of the ship

Further, three velocity components of the Koopman mode decomposition on the previously mentioned $z-$plane is shown in fig. 4.32 together with the isocontour of vorticity colored by the velocity magnitude for the two stable harmonics. Note that frequency is normalized using U_∞ as the reference velocity and funnel width of the ship $L = 3.048$m as the characteristic length scale (*Forrest and Owen*, 2010). As expected, the spKDMD yields a large number of decaying modes with only three nontrivial stable harmonics, since significant transient effects are present in the data. From the Koopman mode decomposition in fig. 4.32, we observe the following:

- modes with eigenvalues close to each other exhibit similar spatial structure,
- modes associated with higher frequency are dominated by smaller scales,
- the stable harmonic mode near $St = 0.09$ associated with a strong cone-shape vortex originating from the upwind (right) rear edge of the superstructure on the ship,

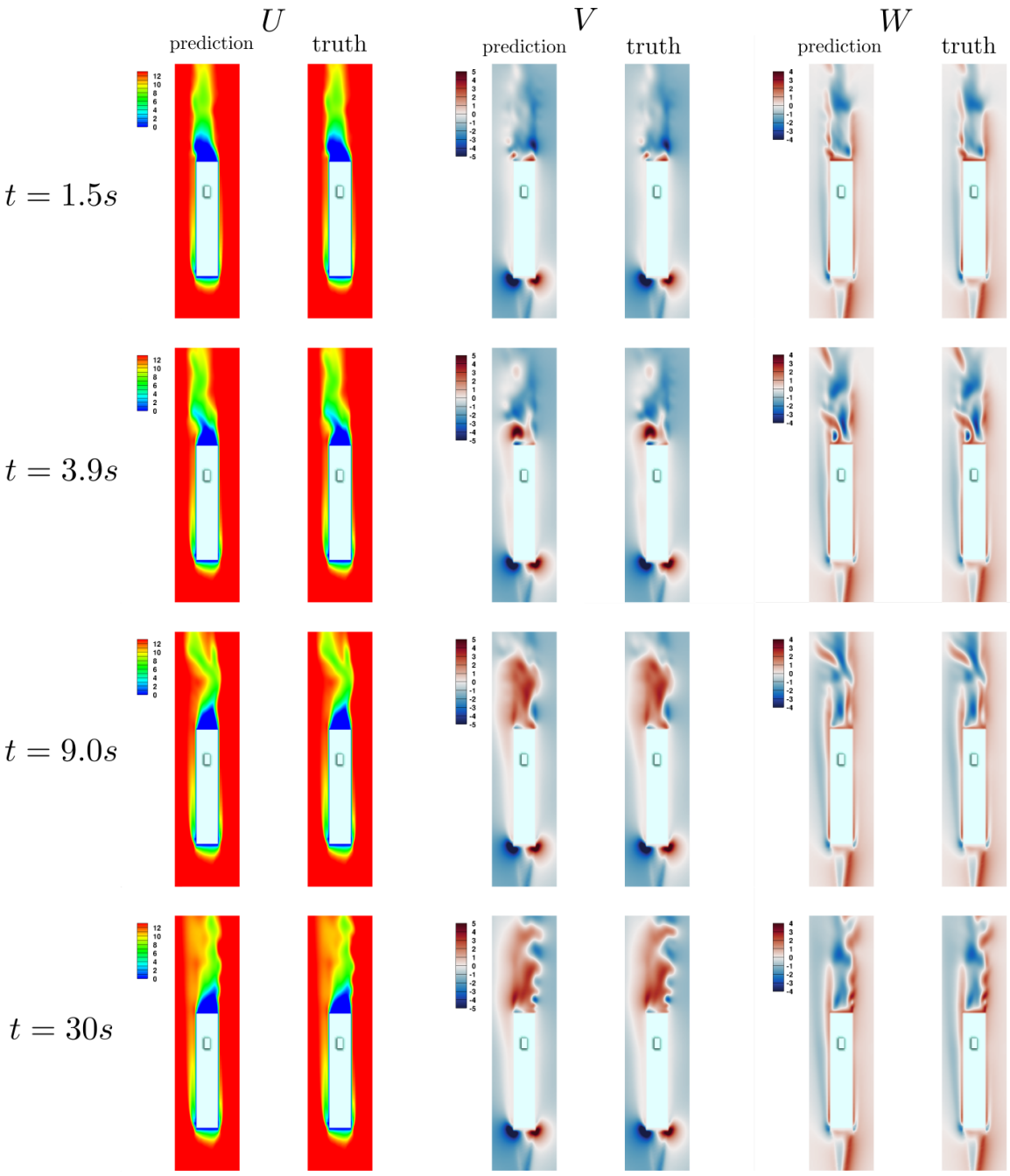


Figure 4.31: Contour of velocity components near the ship on $z-$plane slice at $t = 1.5s$, 3.9s, 9.0s, 30s. For each subfigure, left: prediction from KDMD; right: ground truth.

- the stable harmonic mode near $St = 0.068$ corresponds to vortex shedding induced by the funnel,
- the slowly-decaying mode near $St = 0.022$ shows unsteady circulation directly behind the superstructure,
- the steady mode ($St = 0$) is consistent with the large circulation behind the superstructure, the roll-up wind on the side of landing deck, and vertical suction towards the floor on the landing deck.

4.4.3.2 Comparison with sparsity-promoting DMD

We again repeat the comparison between our spKDMD and spDMD (*Jovanović et al.*, 2014). Note that DMD on the first 40 POD modes performs poorly similar to section 4.4.2.3 and therefore not shown here. To make a fair comparison against the spKDMD from previous subsection, however, we collect the first 200 POD coefficients for spDMD to ensure that a sufficient number of modes are used to fit the trajectory well. We then carefully choose the penalty coefficient in spDMD to ensure that the same number of modes are retained as in spKDMD. As shown in fig. 4.33, within the time horizon $t < 50$, a posteriori evaluation shows that spKDMD offers much improved predictions compared to spDMD (*Jovanović et al.*, 2014) on the testing trajectory.

Moreover, as further illustrated in the left subfigure of fig. 4.34, eigenvalues identified from spKDMD only contain two stable modes while nearly all eigenvalues from spDMD are located near the unit circle, among which there are 30 out of 56 slightly unstable modes. These unstable modes inevitably lead to *the identified system being numerically unstable* when predicting beyond the current training time horizon, whereas spKDMD predicts a "physically consistent" limit cycle behavior. As indicated in the right subfigure of fig. 4.34, such instability is related to the inability

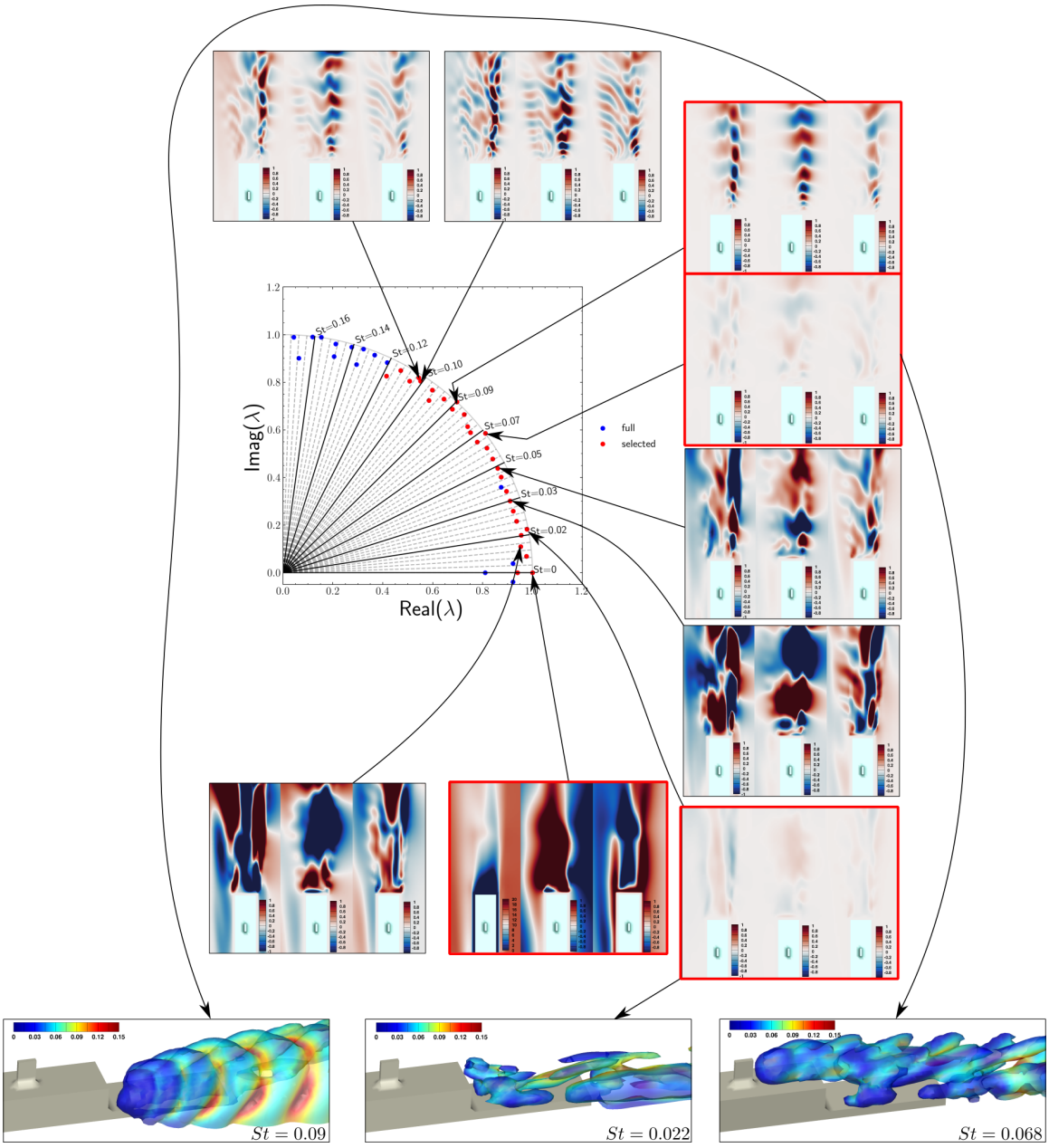


Figure 4.32: Contours of Koopman modes of ship airwake on the z-plane at $t = 0$. For each subfigure, left: U, middle: V, right: W. Red squares indicate stable modes. Bottom: iso-contour of vorticity colored by velocity magnitude zoomed near the landing deck.

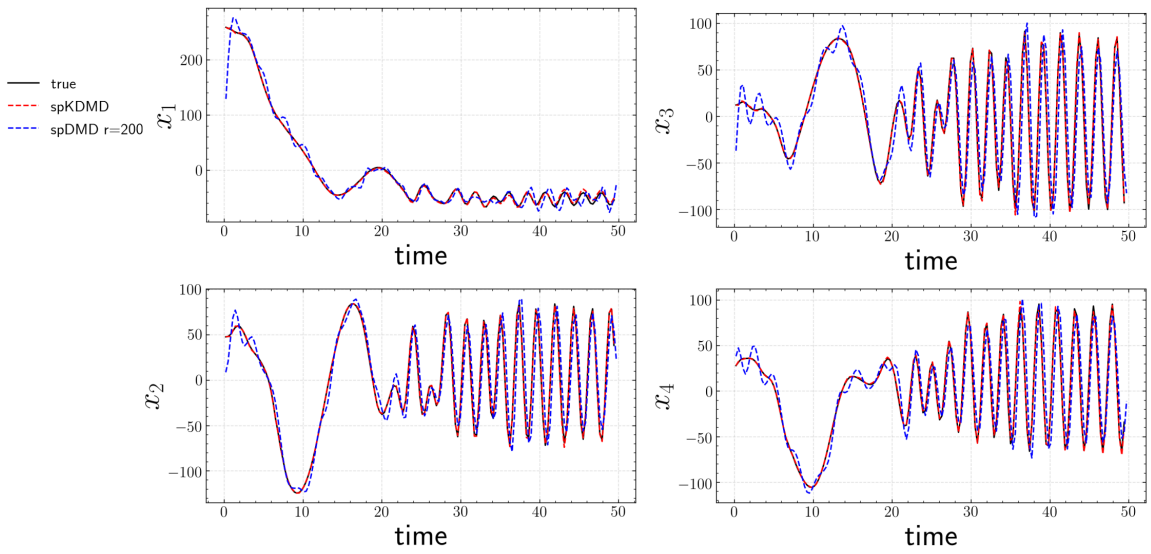


Figure 4.33: Comparison of a posteriori prediction of the 4 most significant POD coefficients of the testing trajectory between sparsity-promoting KDMD and spDMD (*Jovanović et al.*, 2014) for the 3D ship-airwake flow. x_i denotes i-th POD coefficient.

of the (linear) features to approximate the Koopman-invariant subspace, where only 8 modes are within 10% of maximal deviation from linear evolution, compared to 60 modes in KDMD. We note that similar observations of the drastically different eigenvalue distribution were reported in the original KDMD paper (*Williams et al.*, 2014).

4.5 Summary

The main contribution of this chapter is a model agnostic technique to extract compact Koopman-invariant subspaces. Particularly, we developed a) sparsity promoting technique based on a posteriori error analysis, and b) multi-task learning techniques for mode selection as an extension of spDMD into the non-linear variants. Further, analytical relationships between spDMD, Kou's criterion, and the proposed method are derived from the viewpoint of optimization. The algorithm is first evaluated in detail on a simple two state dynamical system, for which the Koopman

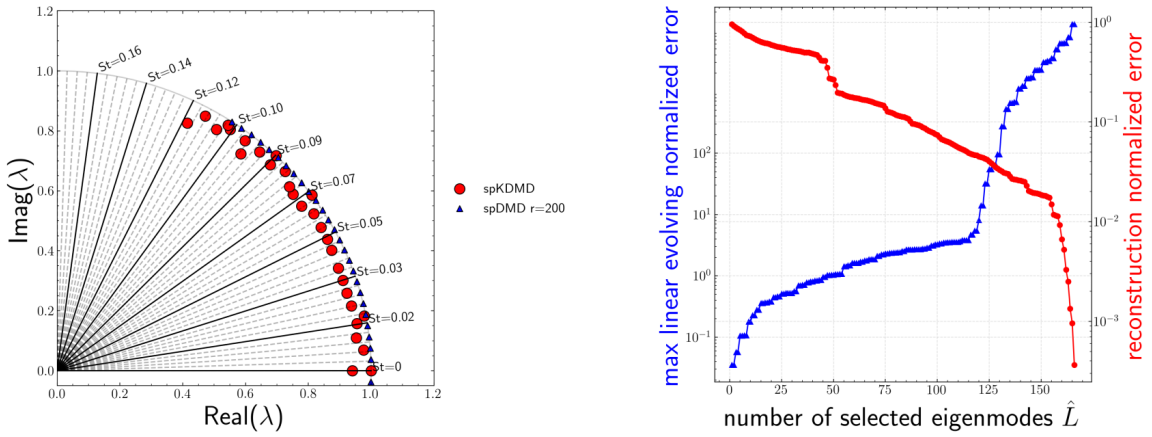


Figure 4.34: Left: Comparison of identified eigenvalues between spKDMD and spDMD (*Jovanović et al.*, 2014) for the 3D ship-airwake flow. Right: Trend of linear evolving error Q and reconstruction error R from DMD for the 3D ship-airwake flow.

decomposition is known analytically. If one is only interested in the post-transient dynamics of the system *on* an attractor, linear observables with time delays are sufficient to extract an informative Koopman-invariant subspace. Thus, the present techniques are evaluated on two unsteady flows which involve strong transients: 2D flow over a cylinder at different Reynolds numbers and a 3D ship air wake. We demonstrate the effectiveness of discovering *accurate* and *informative* Koopman-invariant subspaces from data and constructing accurate reduced-order models from the viewpoint of Koopman theory. Furthermore, with the proposed algorithm, the parametric dependency of Koopman mode shapes on the Reynolds number is investigated for the cylinder flows. In this case, as Re increases from 70 to 130, the shape of stable modes become flattened downstream and larger upstream. Moreover, the similarity of mode shapes between Koopman modes with similar eigenvalues is observed in both fluid flows. Specifically, five clusters of eigenvalues are observed in the case of 2D cylinder wake flow which is confirmed with weakly non-linear theoretical analysis from *Bagheri* (2013). Type-I, II clusters are found to correspond to fluctuations in lift and drag, respectively. We identify non-oscillatory as well as oscillatory cancellations from the

above two clusters with distinct frequencies. These modes could be understood as "oscillatory shift modes", as a generalization of the model proposed by *Noack et al.* (2003). For the 3D ship airwake case, two stable modes, and one slowly-decaying mode with distinct frequencies and mode shapes resulting from vortex shedding are extracted, and accurate predictive performance is observed in the transient regime while spDMD produces a set of unstable modes around the unit circle.

CHAPTER V

Robust Deep Learning for Koopman Operators with Uncertainty Quantification

> And whenever the modelers come in, they give a worst-case scenario and a best-case scenario. Generally, the reality is somewhere in the middle. I've never seen a model of the diseases that I've dealt with where the worst-case scenario actually came out. They always overshoot.
>
> *Dr. Anthony Fauci on CNN's "State of the Union" (2020)*

5.1 Background and Motivation

Recently, several attempts have leveraged deep learning architectures (*Otto and Rowley*, 2019a; *Li et al.*, 2017; *Takeishi et al.*, 2017; *Wehmeyer and Noé*, 2018; *Yeung et al.*, 2017), to extract Koopman decompositions. *Yeung et al.* (2017) used feed-forward neural networks to learn the dictionary function for the Koopman operator,

but the reconstruction loss was not included. *Li et al.* (2017) enforced several non-trainable functions, e.g., components of the system state, in the Koopman observables to ensure an accurate reconstruction but one that could be inefficient in terms of obtaining a finite-dimensional Koopman observable subspace (*Otto and Rowley*, 2019a). Further, *Takeishi et al.* (2017) utilized linear time-delay embedding in the feedforward neural network framework to construct Koopman observables with non-linear reconstruction, which is critical for partially observed systems (*Pan and Duraisamy*, 2018b). *Lusch et al.* (2018a) further extended the deep learning framework to chaotic systems. *Otto and Rowley* (2019a) considered a recurrent loss for better performance on long time prediction on trajectories that transit to the attractor. Recently, *Morton et al.* (2019) addressed the uncertainty in a deep learning model with a focus on control. The benefit of formulating the search for the Koopman operator in an optimization setting enables the enforcement of stability. For example, it is also feasible to constrain eigenvalues in optimized DMD (*Chen et al.*, 2012; *Askham and Kutz*, 2018). Specifically, in the neural network context, *Erichson et al.* (2019) considered stability promoting loss to encourage Lyapunov stability of the dynamical system. From the viewpoint of carefully managing uncertainties, *Galioto and Gorodetsky* (2020) introduced a comprehensive Bayesian framework for DMD and dictionary-based non-linear regression techniques.

A unified approach towards uncertainty quantification, stabilization, and incorporation of physics information is lacking for the Koopman operator. Further, most of these techniques are formulated in a *discrete* dynamical system setting. This motivates us to establish a probabilistic stabilized deep learning framework specifically to learn the Koopman decomposition for a continuous dynamical system. We employ automatic differentiation variational inference (ADVI) (*Kucukelbir et al.*, 2017) to quantify parametric uncertainty in deep neural networks and structural parameterization to enforce stability of the Koopman operator extracted from the deep neural

networks. A broad comparison of the present work with other approaches is shown in table 5.1.

Table 5.1: Summary of comparison with other Koopman approximation techniques in the literature.

Previous works	continuous /discrete	non-linear reconstruction	continuous spectrum	uncertainty	stability
Yeung et al. (2017)	discrete	✗	✗	✗	✗
Li et al. (2017)	discrete	✗	✗	✗	✗
Takeishi et al. (2017)	discrete	✓	✗	✗	✗
Otto and Rowley (2019a)	discrete	✓	✗	✗	✗
Lusch et al. (2018a)	discrete	✓	✓	✗	✗
Morton et al. (2019)	discrete	✓	✗	✓	✗
Erichson et al. (2019)	discrete	✓	✗	✗	✓
Our framework	continuous	✓	✗	✓	✓

In contrast to previous approaches (*Otto and Rowley*, 2019a; *Lusch et al.*, 2018a; *Yeung et al.*, 2017; *Li et al.*, 2017), our framework pursues continuous-time Koopman decompositions. The continuous formulation is more amenable to posit desired constraints and contend with non-uniform sampling (*Otto and Rowley*, 2019a; *Askham and Kutz*, 2018), which is frequently encountered in experiments or temporal multi-scale data. To begin with, recall the general form of autonomous continuous non-linear dynamical systems,

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{M} \subset \mathbb{R}^n. \tag{5.1}$$

We seek a finite dimensional Koopman invariant subspace $\mathcal{F}_D$ with D linearly independent smooth observation functions, defined as

$$\mathcal{F}_D = \text{span}\{\phi_1, \ldots, \phi_D\} \subset \mathcal{F}, \tag{5.2}$$

where $\phi_i \in C^1(\mathcal{M}, \mathbb{R})$ and $\phi_i \in \mathcal{F}$, $i = 1, \ldots, D$. Correspondingly, the observation

vector $\mathbf{\Phi}$ is defined as

$$\mathbf{\Phi}(x) = \begin{bmatrix} \phi_1(x) & \phi_2(x) & \dots & \phi_D(x) \end{bmatrix} \in \mathcal{F}^D. \tag{5.3}$$

Based on the aforementioned structure, we require the following two conditions. First, $\mathbf{\Phi}(x)$ evolves linearly in time, i.e., $\exists\ \mathbf{K} \in \mathbb{R}^{D\times D}$ s.t.

$$\dot{\mathbf{\Phi}} \triangleq d\mathbf{\Phi}/dt = \mathbf{\Phi}\mathbf{K}. \tag{5.4}$$

By the chain rule, the relationship between eq. (5.4) and eq. (5.1) is

$$\mathbf{F} \cdot \nabla_{\mathbf{x}} \mathbf{\Phi} = \mathbf{\Phi}\mathbf{K}. \tag{5.5}$$

Second, $\exists\ \mathbf{\Psi} : \mathbb{R}^D \mapsto \mathcal{M}$, s.t. $\mathbf{\Psi} \circ \mathbf{\Phi} = \mathcal{I}$, $\mathcal{I} : \mathcal{M} \mapsto \mathcal{M}$ is the identity map. Therefore, we can recover the state $\mathbf{x}$ from $\mathbf{\Phi}$.

From now on, we will use deep neural nets to find the Koopman operator. Note that the formulation of deep learning DMD in differential and recurrent forms are previously presented in section 2.5. However, before that, we will highlight a difference between non-linear reconstruction and linear reconstruction.

5.2 Non-linear reconstruction vs linear reconstruction

The original concept of the Koopman operator studies the evolution of any observable in the Koopman invariant subspace. Thus, the observables of interest can be *linearly* reconstructed from the Koopman eigenfunctions. Regarding the fact that *linear reconstruction* is desirable especially in the content of control (*Kaiser et al.*, 2017). *Li et al.* (2017) considered augmenting the Koopman invariant subspace with neural network-trained observables together with the system state $\mathbf{x}$ to force linear reconstruction. However, for systems with multiple fixed points, there does not exist

a finite-dimensional Koopman invariant subspace that also *spans* the state *globally*[1] due to the impossibility of establishing such a topological conjugacy (*Brunton et al.*, 2016a) with the linear system. Therefore, attention naturally moves to the introduction of non-linear reconstruction that is more expressive than linear reconstruction, together with extra modes that indicate different basins of each attractor (*Takeishi et al.*, 2017; *Otto and Rowley*, 2019a). *Otto and Rowley* (2019b) reported that non-linear reconstruction improves prediction accuracy over linear reconstruction on the problem of transient flow past cylinder. Unfortunately, non-linear reconstruction is equivalent to removing the concept of *Koopman modes* (*Otto and Rowley*, 2019a). In other words, this is equivalent to removing the constraint that the system state lies in the finite-dimensional Koopman invariant subspace. Unfortunately, this becomes a disadvantage in the context of modal analysis for fluid flows.

Although systems with multiple fixed points cannot be topological conjugate to a linear system, in section 5.7 we will show that linear reconstruction is still helpful if we relax the topological conjugacy to linearly spanning the system state $\mathbf{x}$ in a weak sense.

5.3 Guaranteed stabilization of the Koopman operator

Eigenvalues of the Koopman operator are critically important in understanding the temporal behavior of certain modes in the dynamical system. For a measure-preserving system (*Budišić et al.*, 2012) or systems on an attractor, e.g., post-transient flow dynamics (*Arbabi and Mezić*, 2017), even if the system is chaotic, the eigenspectrum of the continuous-time Koopman operator would still be on the imaginary axis. It should be noted that although the Koopman operator still accepts unstable modes, i.e., the real part of the eigenvalues of continuous-time Koopman operator being pos-

[1] In this thesis, the term "global embedding" is used to imply non-locality in phase space. Note that the Hartman-Grobman theorem (*Arrowsmith and Place*, 1992) establishes a topological conjugacy to a linear system with the same eigenvalues in a small neighborhood of the fixed point.

itive, its absence in systems governed by Navier-Stokes equation in fluid mechanics has been documented (*Mezić*, 2013). Hence, in this work, we assume that the Koopman eigenvalues corresponding to the finite dimensional Koopman invariant subspace of interest have non-positive real parts. It is important to note that the concept of unstable Koopman modes should not be confused with that of flow instability. Prior models (for instance, *Otto and Rowley* (2019a)) have not explicitly taken stability into account, and thus resulted in slightly unstable Koopman modes. While this is acceptable for relatively short time predictions, long time predictions will be problematic.

Before presenting the stabilization technique, it is instructive to note the non-uniqueness of *ideal* observation functions $\mathbf{\Phi}$, i.e., the one corresponding to the exact Koopman invariant subspace. This is because one can simultaneously multiply any $D \times D$ invertible real matrix and its inverse before and after the observation vector $\mathbf{\Phi}$ while keeping the Koopman eigenfunction, eigenvalues, and the output from the reconstruction the same. Thus, observation functions described by neural networks cannot be expected to be uniquely determined. We will leverage this non-uniqueness to enforce stability.

Enlightened by recent studies (*Haber and Ruthotto*, 2017; *Chang et al.*, 2018, 2019) in the design of stable deep neural network structures where skew-symmetric weights are used inside non-linear activations, we devised a novel parameterization for the realization of the Koopman operator in the following form:

$$\mathbf{K}_{stable} = \begin{bmatrix} -\sigma_1^2 & \zeta_1 & & \\ -\zeta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \zeta_{D-1} \\ & & -\zeta_{D-1} & -\sigma_D^2 \end{bmatrix}, \tag{5.6}$$

where $\zeta_1, \ldots, \zeta_{D-1}$, $\sigma_1, \ldots, \sigma_D \in \mathbb{R}$.

In appendix E, we show that the real parts of the eigenvalues of a $n \times n$ real (possibly non-symmetric) negative semi-definite matrix $\mathbf{A}$ are non-positive. We then prove that the above parameterization posits a constraint that the time evolution associated with $\mathbf{K}_{stable}$ in eq. (5.6) for any choice of parameters in $\mathbf{R}$ is always stable. Further, the constraint from the parameterization in eq. (5.6) is actually rich enough such that any diagonalizable matrix corresponding to a stable Koopman operator can be represented without loss of expressivity.

Theorem V.1. *For any real square diagonalizable matrix $\mathbf{K} \in \mathbb{R}^{D\times D}$ that only has non-positive real parts of the eigenvalues $D \geqslant 2$, there exists a set of $\zeta_1, \ldots, \zeta_{D-1}$, $\sigma_1, \ldots, \sigma_D \in \mathbb{R}$ such that the corresponding $\mathbf{K}_{stable}$ in eq. (5.6) is similar to $\mathbf{K}$ over $\mathbb{R}$. Moreover, for any $\zeta_1, \ldots, \zeta_{D-1}$, $\sigma_1, \ldots, \sigma_D \in \mathbb{R}$, the real part of the eigenvalues of the corresponding $\mathbf{K}_{stable}$ is non-positive.*

Proof. For any real square diagonalizable matrix $\mathbf{K} \in \mathbb{R}^{D\times D}$ that only has non-positive real parts of the eigenvalues, there exists an eigendecomposition,

$$\mathbf{K} = \mathbf{M}\mathbf{J}\mathbf{M}^{-1}, \quad \mathbf{M}, \mathbf{J} \in \mathbb{C}^{D\times D}. \tag{5.7}$$

Without loss of generality, the diagonal matrix $\mathbf{J}$ contains $2D_c$ complex eigenvalues $\{\lambda_j^c\}_{j=1}^{2D_c}$ and D_r real eigenvalues $\{\lambda_j^r\}_{j=1}^{D_r}$ where $2D_c + D_r = D$.

Consider a 2×2 real matrix, $\mathbf{A}_j = \begin{bmatrix} -\sigma_j^2 & \zeta_j \\ -\zeta_j & -\sigma_j^2 \end{bmatrix}$ where the eigenvalues are $\lambda_{1,2} = -\sigma_j^2 \pm j\zeta_j^2$. We use this matrix to construct a 2×2 matrix that has eigenvalues λ_j^c. For each pair of complex eigenvalues, $j = 1, \ldots, D_c$, we have $\mathbf{A}_j = \begin{bmatrix} \mathrm{Re}(\lambda_j^c) & \sqrt{|\mathrm{Im}(\lambda_j^c)|} \\ -\sqrt{|\mathrm{Im}(\lambda_j^c)|} & \mathrm{Re}(\lambda_j^c) \end{bmatrix}$. Next, combining with the D_r real eigenvalues, we have

the following block diagonal matrix that shares the same eigenvalues as $\mathbf{J}$,

$$\widetilde{\mathbf{K}} = \begin{bmatrix} \mathbf{A}_1 & & & & & \\ & \ddots & & & & \\ & & \mathbf{A}_{D_c} & & & \\ & & & \lambda_1^r & & \\ & & & & \ddots & \\ & & & & & \lambda_{D_r}^r \end{bmatrix} \in \mathbb{R}^{D \times D}. \tag{5.8}$$

Since two matrices with the same (complex) eigenvalues are similar (*Meyer*, 2000), $\mathbf{K}$ is similar to $\widetilde{\mathbf{K}}$ over $\mathbb{C}$. However, since $\mathbf{K}$ and $\widetilde{\mathbf{K}}$ are matrices over $\mathbb{R}$, they are already similar over $\mathbb{R}$ (Refer Corollary 1 on p. 312 in Ref. (*Herstein*, 1975)). Notice that the general form of $\widetilde{\mathbf{K}}$ is just a special form of $\mathbf{K}_{stable}$ in eq. (5.6). Therefore, one can always find a set of $\zeta_1, \ldots, \zeta_D$ and $\sigma_1, \ldots, \sigma_D$ such that $\mathbf{K}_{stable}$ is similar to $\mathbf{K}$ over $\mathbb{R}$. Next, notice that given $\sigma_1, \ldots, \sigma_D$, $\zeta_1, \ldots, \zeta_D$, $\forall \mathbf{v} \in \mathbb{R}^{D \times 1}$, we have

$$\mathbf{v}^\top \mathbf{K}_{stable} \mathbf{v} = \frac{1}{2} \mathbf{v}^\top (\mathbf{K}_{stable} + \mathbf{K}_{stable}^\top) \mathbf{v} = \mathbf{v}^\top \begin{bmatrix} -\sigma_1^2 & & \\ & \ddots & \\ & & -\sigma_D^2 \end{bmatrix} \mathbf{v} \leqslant 0. \tag{5.9}$$

Therefore, $\mathbf{K}_{stable}$ is a negative semi-definite matrix. Following lemma E.2, the real part of any eigenvalue of $\mathbf{K}_{stable}$ is non-positive. □

Thanks to the conjugation over $\mathbb{R}$, the last layer of encoder, i.e., $\mathbf{\Phi}$, and the first layer of the decoder, i.e., $\mathbf{\Psi}$, will absorb any linear transformation necessary. Hence, the above theorem simply means one can parameterize $\mathbf{K}$ with eq. (5.6) that guarantees stability of the linear dynamics of $\mathbf{\Phi}$ without loss of expressibility for cases where unstable modes are absent.

Careful readers might notice that indeed one can further truncate the parameter-

ization with a 2×2 block matrix instead of the tridiagonal form used in eq. (5.6). However, since this reduction takes more effort in the implementation while the parameterization in eq. (5.6) has already reduced the number of parameters from $O(D^2)$ to $O(D)$, we prefer the tridiagonal form. For the rest of the work, we will use this parameterization for all of the cases concerned.

5.4 Design of neural network architecture with SVD-DMD embedded

In the previous subsection, we described the deep learning formulation for the Koopman operator as a finite dimensional optimization problem that approximates the constrained variational problem, together with a stable parameterization for $\mathbf{K}$. In this subsection, we will describe the design of our neural network that further embeds the SVD-DMD for differential and recurrent forms, which were presented in sections 2.5.2.1 and 2.5.2.2.

5.4.0.1 Input normalization

As a standard procedure, we consider normalization on the snapshot matrix of state variable $\mathbf{X}$,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_{M-1} \end{bmatrix} \in \mathbb{R}^{M \times n} \tag{5.10}$$

for better training performance on neural networks (*Goodfellow et al.*, 2016). Specifically, we consider Z-normalization shown in eq. (5.11), i.e., subtracting the mean of $\mathbf{x}$ then dividing the standard deviation to obtain $\mathbf{z}$.

$$\mathbf{z} = (\mathbf{x} - \overline{\mathbf{x}})\mathbf{\Lambda}^{-1}, \tag{5.11}$$

where $\overline{\mathbf{x}} = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{x}_m$, $\mathbf{\Lambda} = \text{diag}\{d_1, \ldots, d_N\}$, d_j is the uncorrected standard deviation of j-th component of $\mathbf{x}$, i.e.,

$$d_j = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\mathbf{x}_{m,j} - \overline{\mathbf{x}}_j)^2}, \tag{5.12}$$

where $\mathbf{x}_{m,j}$ is j-th component of m-th data, $j = 1, \ldots, n$.

While such a normalization is helpful for neural network training in most cases, in some cases where the data is a set of POD coefficients, it cannot differentiate between components that could be more significant than others. Therefore, for those cases, we consider a different normalization with the $\mathbf{\Lambda}$ that sets the ratio of standard deviation between components as:

$$\mathbf{\Lambda} = d_{max}\mathbf{I}, \tag{5.13}$$

where $d_{max} = \max_{j=1,\ldots,n} d_j$, and $\mathbf{I}$ is the identity matrix.

5.4.0.2 Embedding with SVD-DMD

Instead of directly using the standard feedforward neural network structure employed in previous works, we embed SVD-DMD (*Schmid*, 2010) into the framework and learn the residual. Recall for the standard SVD-DMD algorithm, given M sequential snapshots in eq. (5.10) uniformly sampled at intervals Δt, one linearly approximates the action of $\mathcal{K}_{\Delta t}$ of the first r dominant SVD modes of *centered* snapshots, i.e., $\mathbf{q} = (\mathbf{x} - \overline{\mathbf{x}})\mathbf{V}_r$. Here, r is empirically chosen as a balance between numerical robustness and reconstruction accuracy, and $\mathbf{V}_r$ are the first r columns from $\mathbf{V}$ of the SVD of centered snapshots,

$$\overline{\mathbf{X}} = \begin{bmatrix} \mathbf{x}_1 - \overline{\mathbf{x}} \\ \vdots \\ \mathbf{x}_M - \overline{\mathbf{x}} \end{bmatrix} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top. \tag{5.14}$$

Then the SVD-DMD operator is simply the matrix $\mathbf{A}$ that minimizes $\sum_{m=0}^{M-2}\|\mathbf{q}_{m+1} - \mathbf{q}_m\mathbf{A}\|$, where $\mathbf{q}_m$ is the corresponding orthogonal projection of $\mathbf{x}_m$.

To embed the above SVD-DMD structure into the neural network, we introduce two modifications. First, we take $r = D$. Since D is arbitrary, if $n < D$, we simply append zero columns in $\mathbf{V}_D$. Second, we would also need to accommodate $\mathbf{z}$ with the input normalization in section 5.4.0.1. Thus, we cast $\mathbf{q} = \mathbf{z}\mathbf{\Lambda}\mathbf{V}_D$. We then have,

$$\mathbf{\Phi}_{svd}(\mathbf{z}) = \mathbf{z}\mathbf{\Lambda}\mathbf{V}_D, \quad \mathbf{\Psi}_{svd}(\mathbf{\Phi}) \triangleq \mathbf{\Phi}\mathbf{V}_D^\top\mathbf{\Lambda}^{-1}, \tag{5.15}$$

$$\mathbf{\Phi}(\mathbf{z}) = \underbrace{\mathbf{\Phi}_{nn}(\mathbf{z})W_{enc,L}}_{\text{non-linear observables}} + \underbrace{\mathbf{\Phi}_{svd}(\mathbf{z})W_{enc,L}}_{\text{linear observables}}, \tag{5.16}$$

$$\mathbf{\Psi}(\mathbf{\Phi}) = \underbrace{\mathbf{\Psi}_{nn}(\mathbf{\Phi})}_{\text{non-linear reconstruction}} + \underbrace{\mathbf{\Psi}_{svd}(\mathbf{\Phi}W_{dec,1})}_{\text{linear reconstruction}}, \tag{5.17}$$

where $\mathbf{\Phi}_{nn} \triangleq \mathbf{\Phi}(\cdot; \mathbf{W}_{\mathbf{\Phi}}\backslash\{W_{enc,L}\})$, $\mathbf{W}_{\mathbf{\Phi}} = \{W_{enc,1}, b_{enc,1}, \ldots, W_{enc,L}\}$, $\mathbf{\Psi}_{nn} \triangleq \mathbf{\Psi}(\cdot; \mathbf{W}_{\mathbf{\Psi}})$, $\mathbf{W}_{\mathbf{\Psi}} = \{W_{dec,1}, \ldots, W_{dec,L}, b_{dec,L}\}$, L is the number of layers for encoder or decoder neural network. The embedding is illustrated in fig. 5.1.

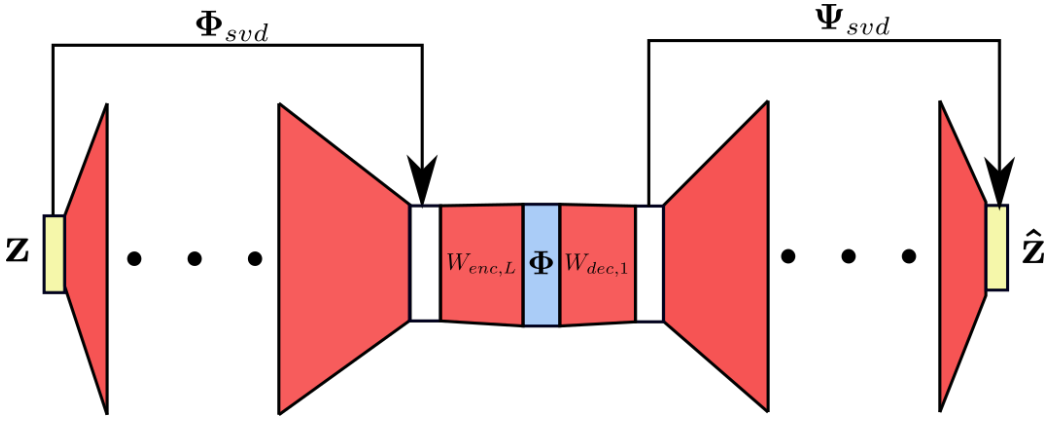


Figure 5.1: Sketch of SVD-DMD embedding in the feedforward neural network. Red blocks represent neural network weights and biases. Yellow blocks represent states. Light blue block represent observables.

The intuition behind the embedding of the SVD-DMD into the framework is given below:

1. Schmid's DMD algorithm (*Schmid*, 2010) relies on the dominant POD modes to reduce the effect of noise, and has been shown to approximate the Koopman invariant subspace (*Rowley et al.*, 2009). This has been demonstrated even for high-dimensional non-linear dynamical systems with millions of degrees of freedom (*Schmid et al.*, 2011). Moreover, for systems with continuous spectra, POD appears to be a robust alternative to Koopman mode decomposition (*Mezić*, 2015). Thus, we assume that the true Koopman invariant subspace is easier to obtain by only learning the residual with respect to the dominant POD subspace.

2. Although we did not precisely implement ResNet blocks (*He et al.*, 2016) (empirically, for the problem concerned in this thesis, our network does not have to be as deep as common architectures in the deep learning community (*He et al.*, 2016)), we believe that such fixed mappings may have similar benefits in ResNets.

3. For an ideal linear dynamical system, using the aforementioned neural network model with non-linear reconstruction can result in an infinite number of unnecessary Koopman modes as global minima. For example, consider the case of a linear dynamical system, i.e., $\mathbf{F}(\mathbf{x}) = \mathbf{x}\mathbf{A}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$. The *desired* Koopman invariant subspace is trivially the span of the projections of $\mathbf{x}$ onto each component and the *desired* Koopman eigenvalues are simply that of $\mathbf{A}$. Assuming simple and real eigenvalues, one can have $\dot{\mathbf{y}} = \mathbf{y}\mathbf{\Lambda}$, where $\mathbf{y} = \mathbf{x}\mathbf{M}$, $\mathbf{A} = \mathbf{M}\mathbf{J}\mathbf{M}^{-1}$ as the eigen-decomposition with $\mathbf{J} = \text{diag}\{\lambda_j\}_{j=1}^{n}$. Then for each component y_j, we have $\dot{y_j} = \lambda_j y_j$. For any $n_j \in \mathbb{N}$, consider the observable $\phi_j = y_j^{2n_j+1}$, one can have $\dot{\phi_j} = (2n_j+1)\lambda_j\phi_j$, i.e., $\text{span}\{\phi_1, \ldots, \phi_N\}$ is invariant to $\mathcal{K}_t$ [2]. Then, consider the non-linear decoder as one that simply takes the

[2] *Budišić et al.* (2012) showed that the set of eigenfunctions naturally forms an Abelian semigroup under pointwise products.

$2n_j + 1$-th root on ϕ_j, and one can recover $\mathbf{y}$ exactly. Finally, augmenting the decoder with $\mathbf{M}^{-1}$ and the encoder with $\mathbf{M}$, the neural network model can find a spurious linear embedding, with eigenvalues as $\{(2n_1+1)\lambda_1, \ldots, (2n_N+1)\lambda_N\}$ rather than the *desired* $\{\lambda_1, \ldots, \lambda_N\}$, which is an over-complicated non-linear reconstruction. It is trivial to generalize the above thought experiment to cases where eigenvalues are complex for a real-input-real-output neural network to accommodate. On the other hand, since most often the neural network is initialized with small weights near zero, the effect of the non-linear encoder and decoder can be small initially compared to the DMD part. Thus, if the system can be exactly represented by DMD, the optimization for the embedded architecture is initialized near the *desired* minimum. We note that this could lead to attenuation of the spurious modes due to the non-linear reconstruction for essentially linear dynamics.[3]

The neural network architecture for the differential form is shown in fig. 5.2, and for the recurrent form in fig. 5.3. Note that, if the non-linear part, i.e., the

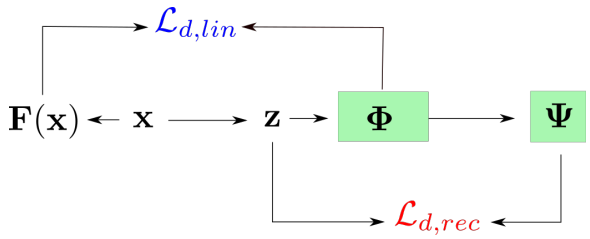


Figure 5.2: Sketch of the framework of learning a continuous-time Koopman operator in the differential form.

feedforward neural network is not activated, the above formulation reverts to an over-parameterized SVD-DMD. Specifically, the recurrent form model with neural network deactivated can be viewed as a simplified version of optimized DMD (*Askham and Kutz*, 2018; *Chen et al.*, 2012).

[3]It has to be mentioned that that such an issue could exist also in cases where DMD is not desired.

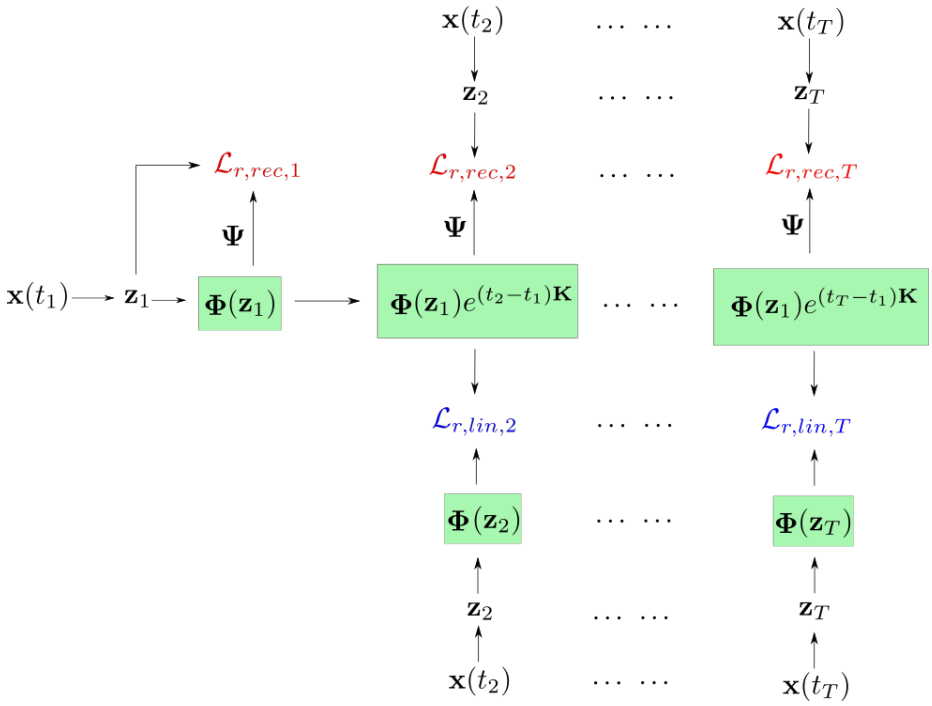


Figure 5.3: Sketch of the framework of learning a continuous-time Koopman operator in the recurrent form.

5.5 Implementation

The framework is built using Tensorflow (*Abadi et al.*, 2016). Neural network parameters $\mathbf{W}_{\mathbf{\Phi}}, \mathbf{W}_{\mathbf{\Psi}}$, are initialized with the standard truncated normal distribution. $\mathbf{K}$ is initialized with the corresponding DMD approximation. The objective function is optimized using Adam (*Kingma and Ba*, 2014), which is an adaptive first order stochastic optimization method using gradient updates scaled by square roots of exponential moving averages of previous squared gradients. Note that we also include weight decay regularization, $\mathcal{L}_{reg} = \|\boldsymbol{W}_{\mathbf{\Phi}}\|^2 + \|\boldsymbol{W}_{\mathbf{\Psi}}\|^2$, in the objective function, to avoid spurious oscillations in the learned Koopman functions, which helps generalization in an interpolation sense (*Goodfellow et al.*, 2016).

5.6 Probabilistic formulation

5.6.1 Bayesian neural networks

A Bayesian formalism is adopted to quantify the impact of several sources of uncertainty in model construction on the model predictions. Bayes' rule is

$$P(\boldsymbol{\Theta}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})}{P(\mathcal{D})} \iff \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}, \tag{5.18}$$

where $\mathcal{D}$ is data, and $\boldsymbol{\Theta}$ is the set of parameters. For simplicity, P represents the probability density function (PDF) on the measure space generated by the data and parameters. From a traditional Bayesian standpoint, as the number of parameters in the neural network is large, it is impossible for common inference tools such as Markov Chain Monte Carlo (MCMC) to be practical. To overcome the curse of dimensionality, several general approaches such as Laplacian approximation (*Denker and Lecun*, 1991) and variational inference (*Kucukelbir et al.*, 2017; *Blundell et al.*, 2015) have been proposed. The former is computationally economical but has two major limitations: First, computing the full Hessian is impossible and expensive for a high dimensional problem and most often approximations such as $\mathcal{J}^\top\mathcal{J}$, where $\mathcal{J}$ is the Jacobian are employed *MacKay* (1992). Second, it only provides a local approximation, which can often be far removed from the true posterior. Variational inference has become popular in the deep learning community as it offers an informed balance between the computationally expensive MCMC method, and the cheap but less descriptive models such as the Laplacian approximation. Historically, variational inference for neural networks (*Hinton and Van Camp*, 1993) has been difficult (*Neal*, 2012) to formulate, largely due to the difficulty of deriving analytical solutions to the required integrals over the variational posteriors (*Graves*, 2011) even for simple network structures. *Graves* (2011) proposed a stochastic method for variational inference with a diagonal Gaussian posterior that can be applied to almost any dif-

ferentiable log-loss parametric model, including neural networks. However, there is always a trade-off between the complexity of the posterior and scalability and robustness (*Zhang et al.*, 2018). In this work, we adopt the mean-field variational inference (*Kucukelbir et al.*, 2017).

5.6.2 Mean-field variational inference

As illustrated in the left figure of fig. 5.4, the key idea in variational inference (*Blei et al.*, 2017) is to recast Bayesian inference as an optimization problem by searching the best parameterized probability density function $q(\boldsymbol{\Theta}; \hat{\xi})$ in a family of approximating densities, namely the variational posterior, $\{q(\boldsymbol{\Theta}; \xi) | \xi \in \Xi\}$, such that it is closest to the true posterior $P(\boldsymbol{\Theta}|\mathcal{D})$. Most often, the Kullback–Leibler (KL) divergence is employed to measure the distance, which is defined as $\mathrm{KL}(q(\boldsymbol{\Theta}; \xi) \| P(\boldsymbol{\Theta}|\mathcal{D})) = \int_\Omega q(\boldsymbol{\Theta}; \xi) \log \frac{q(\boldsymbol{\Theta}; \xi)}{P(\boldsymbol{\Theta}|\mathcal{D})} d\boldsymbol{\Theta} = \mathbb{E}_{q(\boldsymbol{\Theta}; \xi)}\big[\log q(\boldsymbol{\Theta}; \xi) P(\boldsymbol{\Theta}|\mathcal{D})\big]$, where Ω is the support of $q(\boldsymbol{\Theta}; \xi)$. This implies $\mathrm{supp}(q(\boldsymbol{\Theta}; \xi)) \subseteq \mathrm{supp}(P(\boldsymbol{\Theta}|\mathcal{D}))$. Further, we assume $\mathrm{supp}(P(\boldsymbol{\Theta}|\mathcal{D})) = \mathrm{supp}(P(\boldsymbol{\Theta}))$. Ξ is the domain of ξ, depending on the parameterization and family of approximating densities.

However, direct computation of the KL divergence is intractable, since we do not have access to $\log P(\mathcal{D})$. Instead, we choose an alternative, the evidence lower bound (ELBO), i.e., the negative KL divergence plus $\log P(\mathcal{D})$, in eq. (5.19) to be maximized.

$$\mathcal{L}_{elbo}(\xi) = \mathbb{E}_{q(\boldsymbol{\Theta}; \xi)}\big[\log P(\mathcal{D}, \boldsymbol{\Theta})\big] - \mathbb{E}_{q(\boldsymbol{\Theta}; \xi)}\big[q(\boldsymbol{\Theta}; \xi)\big]. \tag{5.19}$$

To maximize the ELBO, we leverage automatic differentiation from Tensorflow to compute the gradients with respect to ξ, following the framework of Automatic Differentiation Variational Inference (ADVI) (*Kucukelbir et al.*, 2017) where Gaussian distributions are considered as the variational family. Specifically, we employ the

mean-field assumption in eq. (5.20) such that,

$$q(\mathbf{\Theta};\xi) = \prod_{j=1}^{Z} q(\theta_j;\xi_j), \tag{5.20}$$

where Z is the total number of parameters. For $j = 1,\ldots,Z$, θ_j is the j-th parameter as random variable, and ξ_j is the corresponding variational parameter that describes the distribution. This is particularly convenient for neural network models constructed in Tensorflow since weights and biases are naturally defined on some real coordinate space. If the support of the parameter distribution is restricted, one can simply consider a one-to-one differentiable coordinate transformation $\Upsilon(\mathbf{\Theta}) = \mathbf{Z}$, such that $\mathbf{Z}$ is not restricted in some real coordinate space, and posit a Gaussian distribution on $\mathbf{Z}$. Note that this naturally induces non-Gaussian distribution. Here, we employ the ADVI functionality in Edward (*Tran et al.*, 2016), which is built upon Tensorflow to implement ADVI. Interested readers should refer to the original paper of ADVI (*Kucukelbir et al.*, 2017) for the specific details of implementing mean-field variational inference including the usage of the reparameterization-trick to compute the gradients. In contrast, note that the maximum a posteriori (MAP) estimation of the posterior can be cast as a regularized deterministic model as illustrated in fig. 5.4. Since weight decay is employed in previous deep learning models (*Otto and Rowley*, 2019a; *Lusch et al.*, 2018a) to learn the Koopman decomposition, one can show that the previous model is essentially a MAP estimation of the corresponding posterior.

Note that the mean-field Gaussian assumption is still simplified, yet effective and scalable for deep neural nets (*Blundell et al.*, 2015). It is interesting to note that several recent works (*Zhu and Zabaras*, 2018; *Zhu et al.*, 2019) leverage Stein Variational Gradient Descent (SVGD) (*Liu and Wang*, 2016), a non-parametric, particle-based inference method, which is able to capture multi-modal posteriors. Robustness and scalability to high dimensions, e.g., deep neural nets, is still an area of active re-

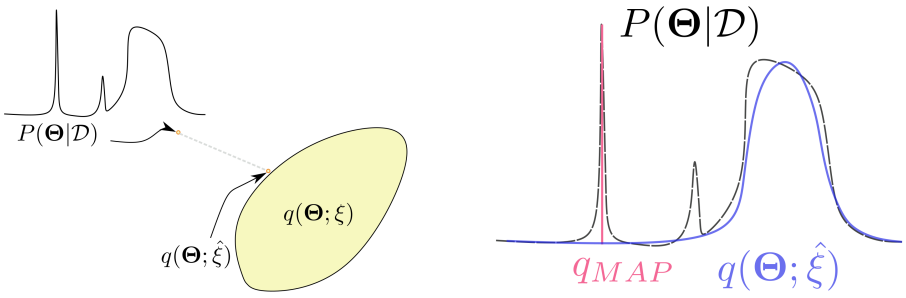


Figure 5.4: Left: illustration of variational inference. Right: difference between MAP and variational inference.

search (*Zhuo et al.*, 2017; *Liu and Wang*, 2018; *Shi et al.*, 2018; *Lu et al.*, 2019). We provide further background on variational inference in appendix A.

5.6.3 Bayesian hierarchical model setup

Recall that we have the following parameters for the deep learning models introduced in section 2.5.2.1 and section 2.5.2.2: 1) weights and biases for the "encoder", $\mathbf{W}_{\mathbf{\Phi}}$, 2) weights and biases for the "decoder", $\mathbf{W}_{\mathbf{\Psi}}$, and 3) stabilized $\mathbf{K}$ with $\zeta_1, \ldots, \zeta_D$ and $\sigma_1, \ldots, \sigma_D$. Based on mean-field assumptions, we just need to prescribe the prior and variational posterior for each parameter. For each weight and bias, we posit a Gaussian prior with zero mean, with the scaling associated with each parameter to follow the recommended half–Cauchy distribution (*Gelman*, 2004; *Polson and Scott*, 2012), which has zero mean and scale as 1 (empirical). The variational posterior for each weight and bias is Gaussian, and Log-normal for scale parameters. For the off-diagonal part of $\mathbf{K}$, ζ_i, we posit the same type of Gaussian prior as before with the scale parameter following a hierarchical model for each $i = 1, \ldots, D$. The corresponding variational posterior for ζ_i is still Gaussian while log-normal for the scale parameter. For the non-negative diagonal part of $\mathbf{K}$, we posit a Gamma distribution for each σ_i^2, $i = 1, \ldots, D$, with rate parameter as 0.5 and shape parameter following the previous half-Cauchy distribution. Variational posteriors for both σ_i^2 and its

shape parameter are log-normal.

For the differential form in section 2.5.2.1, we set up the following likelihood function[4] based on Z-normalized data $\mathcal{D} = \{\mathbf{z}_m, \dot{\mathbf{z}}_m\}_{m=0}^{M-1}$:

$$\mathcal{D}|\boldsymbol{\Theta} \sim \prod_{m=0}^{M-1} \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\Psi}(\boldsymbol{\Phi}(\mathbf{z}_m; \mathbf{W}_{\boldsymbol{\Phi}}); \mathbf{W}_{\boldsymbol{\Psi}}) - \mathbf{z}_m \\ \dot{\mathbf{z}}_m \cdot \nabla_{\mathbf{z}} \boldsymbol{\Phi}(\mathbf{z}_m; \mathbf{W}_{\boldsymbol{\Phi}}) - \boldsymbol{\Phi}(\mathbf{z}_m; \mathbf{W}_{\boldsymbol{\Phi}}) \mathbf{K}_{stable} \end{bmatrix}; \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda}_{rec} & \\ & \boldsymbol{\Lambda}_{lin} \end{bmatrix}\right), \tag{5.21}$$

where $\boldsymbol{\Lambda}_{rec}, \boldsymbol{\Lambda}_{lin}$ are diagonal covariance matrices with the prior of each diagonal element following the previous half–Cauchy distribution. We also posit log-normals to infer the posterior of $\boldsymbol{\Lambda}_{rec}, \boldsymbol{\Lambda}_{lin}$. We denote $\boldsymbol{\Theta}$ as $\mathbf{W}_{\boldsymbol{\Phi}}, \mathbf{W}_{\boldsymbol{\Psi}}$, and associated scale parameters together with $\boldsymbol{\Lambda}_{rec}$, $\boldsymbol{\Lambda}_{lin}$.

For the recurrent form in section 2.5.2.2, given normalized data,

$$\mathcal{D} = \{\{\{\mathbf{z}_{m,j}\}_{j=1}^{T_m}\}_{m=0}^{M-1}, \{\{t_{m,j}\}_{j=1}^{T_m}\}_{m=0}^{M-1}\}, \tag{5.22}$$

we consider the following likelihood, [5]:

$$\mathcal{D}|\boldsymbol{\Theta} \sim \prod_{m=1}^{M} \prod_{j=1}^{T_m} \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\Psi}(\boldsymbol{\Phi}(\mathbf{z}_{m,1}; \mathbf{W}_{\boldsymbol{\Phi}}) e^{t_{m,j} \mathbf{K}_{stable}}; \mathbf{W}_{\boldsymbol{\Psi}}) - \mathbf{z}_{m,j} \\ \boldsymbol{\Phi}(\mathbf{z}_{m,1}; \mathbf{W}_{\boldsymbol{\Phi}}) e^{t_{m,j} \mathbf{K}_{stable}} - \boldsymbol{\Phi}(\mathbf{z}_{m,j}; \mathbf{W}_{\boldsymbol{\Phi}}) \end{bmatrix}; \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda}_{rec} & \\ & \boldsymbol{\Lambda}_{lin} \end{bmatrix}\right). \tag{5.23}$$

5.6.4 Propagation of uncertainties

Given data $\mathcal{D}$ and the inferred posterior $P(\boldsymbol{\Theta}|\mathcal{D})$, assuming a noise-free initial condition $\mathbf{z}_0$, we are interested in future state predictions with uncertainties. For the

[4]Note that independence between the data and the structure of the aleatoric uncertainty is assumed. Such an assumption correlates well with existing deterministic models based on mean-square-error.

[5]Alternative likelihoods can be chosen to account for the variation of aleatoric noise in time, which is well-suited for short-horizon forecasting. However, we are more interested in a free-run situation where only the initial condition is given.

differential form in eq. (5.21), we have

$$
\begin{aligned}
P(\mathbf{z}(t)|\mathbf{z}_0,\mathcal{D}) &= \int P(\mathbf{z}_t|\mathbf{W}_{\mathbf{\Phi}},\mathcal{D},\mathbf{z}_0)P(\mathbf{W}_{\mathbf{\Phi}}|\mathcal{D}))d\mathbf{W}_{\mathbf{\Phi}}, \qquad (5.24)\\
&= \int P(\mathbf{z}_t|\mathbf{\Phi}(\mathbf{z}_0;\mathbf{W}_{\mathbf{\Phi}}),\mathbf{W}_{\mathbf{\Phi}},\mathcal{D},\mathbf{z}_0)P(\mathbf{W}_{\mathbf{\Phi}}|\mathcal{D}))d\mathbf{W}_{\mathbf{\Phi}},\\
&= \iint P(\mathbf{z}_t|\mathbf{\Phi}(t),\mathbf{W}_{\mathbf{\Phi}},\mathcal{D},\mathbf{z}_0)P(\mathbf{\Phi}(t)|\mathbf{\Phi}(\mathbf{z}_0;\mathbf{W}_{\mathbf{\Phi}}),\mathbf{W}_{\mathbf{\Phi}},\mathcal{D},\mathbf{z}_0)P(\mathbf{W}_{\mathbf{\Phi}}|\mathcal{D}))d\mathbf{\Phi}(t)d\mathbf{W}_{\mathbf{\Phi}},\\
&= \iiint P(\mathbf{z}_t|\mathbf{W}_{\mathbf{\Psi}},\mathbf{\Phi}(t))P(\mathbf{\Phi}(t)|\mathbf{\Phi}(\mathbf{z}_0;\mathbf{W}_{\mathbf{\Phi}}))P(\mathbf{W}_{\mathbf{\Phi}}|\mathcal{D}))P(\mathbf{W}_{\mathbf{\Psi}}|\mathcal{D})d\mathbf{\Phi}(t)d\mathbf{W}_{\mathbf{\Psi}}d\mathbf{W}_{\mathbf{\Phi}},\\
&= \iiiint\!\!\iiint P(\mathbf{z}_t|\mathbf{\Lambda}_{rec},\mathbf{W}_{\mathbf{\Psi}},\mathbf{\Phi}(t))P(\mathbf{\Phi}(t)|\mathbf{\Lambda}_{lin},\mathbf{K},\mathbf{\Phi}(\mathbf{z}_0;\mathbf{W}_{\mathbf{\Phi}}))P(\mathbf{W}_{\mathbf{\Phi}}|\mathcal{D}))P(\mathbf{K}|\mathcal{D})\\
&\qquad P(\mathbf{W}_{\mathbf{\Psi}}|\mathcal{D})P(\mathbf{\Lambda}_{lin}|\mathcal{D})P(\mathbf{\Lambda}_{rec}|\mathcal{D})d\mathbf{\Lambda}_{lin}d\mathbf{\Lambda}_{rec}d\mathbf{\Phi}(t)d\mathbf{W}_{\mathbf{\Psi}}d\mathbf{W}_{\mathbf{\Phi}}d\mathbf{K}.
\end{aligned}
$$

However, $P(\mathbf{\Phi}(t)|\mathbf{K},\mathbf{\Phi}(\mathbf{z}_0;\mathbf{W}_{\mathbf{\Phi}}))$ is unknown because the differential form does not use trajectory information. If we assume multivariate Gaussian white noise with the same covariance in the linear loss $\mathbf{\Lambda}_{lin}$, then one can forward propagate the aleatoric uncertainty associated with the likelihood function of the linear loss. Then one immediately recognizes that the continuous-time random process of $\mathbf{\Phi}(t)$ becomes a multivariate Ornstein–Uhlenbeck process,

$$
d\mathbf{\Phi}^{\top}(t) = \mathbf{K}^{\top}\mathbf{\Phi}^{\top}(t)dt + \mathbf{\Lambda}_{lin}^{1/2}d\mathbf{B}(t), \qquad (5.25)
$$

where $\mathbf{B}(t)$ is a D-dimensional Gaussian white noise vector with unit variance. Note that (*Ross et al.*, 1996)

$$
\mathbf{\Phi}(t)|\mathbf{z}_0,\mathbf{\Lambda}_{lin},\mathbf{W}_{\mathbf{\Phi}},\mathbf{K} \sim \mathcal{N}(\mathbf{\Phi}(\mathbf{z}_0;\mathbf{W}_{\mathbf{\Phi}})e^{t\mathbf{K}}, \int_0^t e^{s\mathbf{K}}\mathbf{\Lambda}_{lin}e^{s\mathbf{K}^{\top}}ds), \qquad (5.26)
$$

where $\int_0^t e^{s\mathbf{K}}\mathbf{\Lambda}_{lin}e^{s\mathbf{K}^{\top}}ds = \mathbf{vec}^{-1}(-(\mathbf{K}\oplus\mathbf{K})^{-1}(\mathbf{I}-e^{t(\mathbf{K}\oplus\mathbf{K})})\mathbf{vec}(\mathbf{\Lambda}_{lin}))$. $\mathbf{vec}(\cdot)$ is the stack operator, and $\oplus$ is the Kronecker sum (*Meucci*, 2009).

It is interesting to note that, since $\mathbf{K}$ is restricted by eq. (5.6) and does not contain any eigenvalues with positive real part, the variance in eq. (5.26) will not diverge in

finite time. One can thus simply draw samples of $\mathbf{\Phi}(t)$ from eq. (5.26). Thus, we approximate eq. (5.24) with Monte Carlo sampling from the corresponding variational posterior,

$$P(\mathbf{z}(t)|\mathbf{z}_0, \mathcal{D}) \approx \frac{1}{N_{mc}M_{mc}} \sum_{i=1}^{N_{mc}} \sum_{j=1}^{M_{mc}} P(\mathbf{z}_t|\mathbf{\Lambda}_{lin}^{(i)}, \mathbf{\Lambda}_{rec}^{(i)}, \mathbf{W}_{\mathbf{\Phi}}^{(i)}, \mathbf{K}^{(i)}, \mathbf{W}_{\mathbf{\Psi}}^{(i)}, \mathbf{\Phi}^{(j)}(t), \mathbf{z}_0), \tag{5.27}$$

where the superscript with parentheses represents the index of samples, N_{mc}, M_{mc} are the number of samples corresponding to variational posteriors and the Ornstein–Uhlenbeck process.

For the recurrent form in eq. (5.23), the posterior predictive distribution of $\mathbf{z}(t)$ given the initial condition is straightforward:

$$P(\mathbf{z}(t)|\mathbf{z}_0, \mathcal{D}) = \iiiint P(\mathbf{z}(t)|\mathbf{\Lambda}_{rec}, \mathbf{K}, \mathbf{W}_{\mathbf{\Psi}}, \mathbf{W}_{\mathbf{\Phi}}, \mathbf{z}_0) P(\mathbf{W}_{\mathbf{\Phi}}|\mathcal{D}) P(\mathbf{W}_{\mathbf{\Psi}}|\mathcal{D}) P(\mathbf{K}|\mathcal{D}) \tag{5.28}$$

$$P(\mathbf{\Lambda}_{rec}|\mathcal{D}) d\mathbf{W}_{\mathbf{\Psi}} d\mathbf{\Lambda}_{rec} d\mathbf{K} d\mathbf{W}_{\mathbf{\Phi}},$$

$$\approx \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} P(\mathbf{z}(t)|\mathbf{\Lambda}_{rec}^{(i)}, \mathbf{K}^{(i)}, \mathbf{W}_{\mathbf{\Psi}}^{(i)}, \mathbf{W}_{\mathbf{\Phi}}^{(i)}, \mathbf{z}_0).$$

5.7 Discussion: Role of lifting and linear reconstruction for system with multiple attractors

Brunton et al. (2016a) pointed out that no system with multiple fixed points can admit a finite-dimensional Koopman invariant subspace containing the state variables explicitly. It is tempting to believe that for a system with multiple fixed points, it will be impossible to find a finite-dimensional Koopman invariant subspace to recover the state variables via linear reconstruction. However, many data-driven studies (*Li et al.*, 2017; *Otto and Rowley*, 2019a; *Williams et al.*, 2015) have shown success with linear reconstruction for the above Duffing problem. Surprisingly, *Otto and Rowley* (2019a)

found that linear reconstruction generalizes better than non-linear reconstruction for the Duffing case. Therefore, there appears to be a contradiction between the common knowledge in the dynamical system and is observed empirically.

Here, we provide an explanation hoping to shed light on the above dilemma. *Otto and Rowley* (2019a) empirically found that in the Duffing system, lifting the two-dimensional state into a three-dimensional space, leads to an interesting non-trivial Koopman eigenfunction with a "trivial" Koopman eigenvalue, i.e., zero, acting as an indicator to distinguish different attractors. The idea of partitioning the invariant sets by such Koopman eigenfunctions has been reported before for measure-preserving systems and hybrid systems (*Govindarajan et al.*, 2016; *Mezić and Wiggins*, 1999). As shown in fig. 5.5, the third "indicator" Koopman eigenfunction shows that the neural network has learned how to differentiate the basins of attraction, i.e., initial conditions that would lead to different long-term behavior, by letting them evolve on "parallel" subspaces.

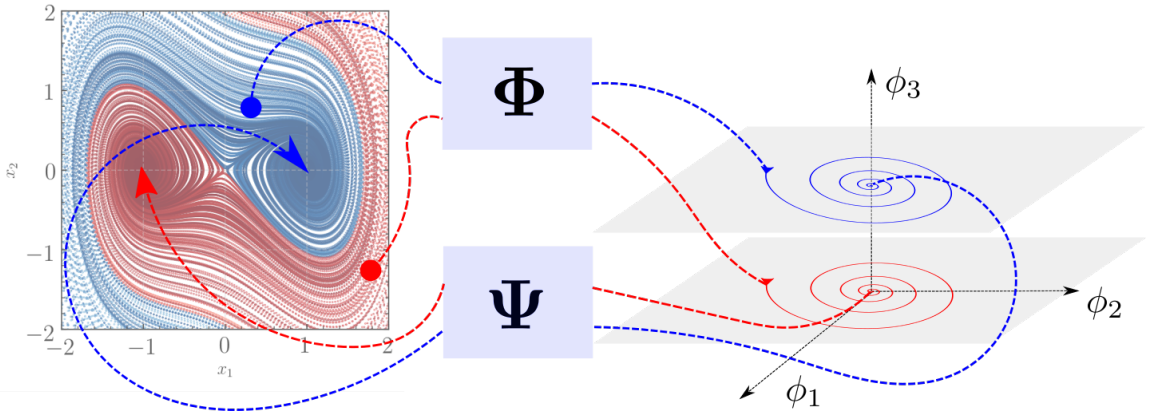


Figure 5.5: Illustration of lifting a two-dimensional Duffing oscillator into three-dimensional space. Note that the observables ϕ_1, ϕ_2, ϕ_3 in the figure is only for the ease of illustration since it is just *one* of the acceptable observables rather than the *one* we obtained in the actual training.

We then consider the two following assumptions:

1. For the system in eq. (5.1), there are J attractors on $\mathcal{M}$, $J \in \mathbb{N}$. Moreover, for

$j = 1, \ldots, J$, the union of disjoint basins of attraction is $\Sigma = \bigcup_{j=1}^{J} \mathcal{S}_j \subset \mathcal{M}$ where $\mathcal{S}_j \subset \mathcal{M}$, is full measure[6].

2. For $j = 1, \ldots, J$, within j-th basin of attraction, one can always find a finite-dimensional Koopman invariant subspace from real functions[7], i.e., $\widetilde{\mathcal{F}}_{D_j} = \text{span}\{\widetilde{\phi}_{j,1}, \ldots, \widetilde{\phi}_{j,D_j}\}$, with $D_j \geqslant n$, that *spans* the *centered* state variable $\widetilde{\mathbf{x}} = \mathbf{x} - \mathbf{b}_j$ locally in $\mathcal{S}_j$,

$$\forall \mathbf{x} \in \mathcal{S}_j, \quad \sum_{l=1}^{D_j} \mathbf{w}_{j,l} \widetilde{\phi_{j,l}}(\widetilde{\mathbf{x}}) = \widetilde{\mathbf{x}}. \tag{5.29}$$

Equivalently in the uncentered coordinate space, for j-th basin of attraction, with $\phi_{j,l}(\mathbf{x}) = \widetilde{\phi_{j,l}}(\mathbf{x} - \mathbf{b}_j)$, one has $\mathcal{F}_{D_j} = \text{span}\{\phi_{j,1}, \ldots, \phi_{j,D_j}\}$, that *linearly reconstructs* the state variable $\mathbf{x}$,

$$\forall \mathbf{x} \in \mathcal{S}_j, \quad \mathbf{b}_j + \sum_{l=1}^{D_j} \mathbf{w}_{j,l} \phi_{j,l}(\mathbf{x}) = \mathbf{x}, \tag{5.30}$$

where $\mathbf{b}_j$ is some constant vector associated to the attractor, e.g., fixed point location, and $\mathbf{w}_{j,l} \in \mathbb{R}^{n \times 1}$ and $\text{rank}(\begin{bmatrix} \mathbf{w}_{j,1} & \ldots & \mathbf{w}_{j,D_j} \end{bmatrix}) = n$.

Now consider the following observable vector $\mathbf{\Phi}$ in $\mathcal{F}^D$, where $D = J + \sum_{j=1}^{J} D_j$,

$$\mathbf{\Phi} = \begin{bmatrix} \chi_{\mathcal{S}_1} & \chi_{\mathcal{S}_1}\phi_{1,1} & \cdots & \chi_{\mathcal{S}_1}\phi_{1,D_1} & \cdots & \chi_{\mathcal{S}_J} & \chi_{\mathcal{S}_J}\phi_{J,1} & \cdots & \chi_{\mathcal{S}_J}\phi_{J,D_J} \end{bmatrix}, \tag{5.31}$$

where $\chi_{\mathcal{S}}(\mathbf{x}) : \mathcal{M} \mapsto \{0, 1\}$ is the indicator function defined as,

$$\chi_{\mathcal{S}}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.32}$$

[6]Clearly, not all continuous dynamical systems belong to such a category. An alternative partitioning for a measure-preserving continuous system is the ergodic partition (*Mezić and Wiggins*, 1999). Due to Birkhoff's pointwise ergodic theorem, one can guarantee that such partitions can cover almost the entire manifold.

[7]We restricted to real functions due to the its popularity in deep learning.

Note that $\chi_{\mathcal{S}_j}$ is also invariant to $\mathcal{K}_t$.

Then we can find a linear combination of each component of $\mathbf{\Phi}$, namely $\widehat{\mathbf{x}} = \sum_{j=1}^{J} \chi_{\mathcal{S}_j}(\mathbf{b}_j + \sum_{l=1}^{D_j} \mathbf{w}_{j,l}\phi_{j,l})$, such that it approximates the state variable $\mathbf{x}$ almost everywhere,

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_{\mathcal{F}}^2 = \left\|\mathbf{x} - \sum_{j=1}^{J} \chi_{\mathcal{S}_j}(\mathbf{b}_j + \sum_{l=1}^{D_j} \mathbf{w}_{j,l}\phi_{j,l})\right\|_{\mathcal{F}}^2 = \int_{\mathcal{M}} \left\|\mathbf{x} - \sum_{j=1}^{J} \chi_{\mathcal{S}_j}(\mathbf{b}_j + \sum_{l=1}^{D_j} \mathbf{w}_{j,l}\phi_{j,l})\right\|^2 d\mu \tag{5.33}$$

$$= \sum_{k=1} \int_{\mathcal{S}_k} \left\|\mathbf{x} - \mathbf{b}_j - \sum_{l=1}^{D_j} \mathbf{w}_{k,l}\phi_{k,l}\right\|^2 d\mu = 0.$$

Back to the apparent dilemma, although it is impossible to find a finite-dimensional Koopman invariant subspace that globally spans the state, as indicated by both numerical experiments (*Otto and Rowley*, 2019a; *Li et al.*, 2017) and analysis above, it is possible in a weak sense, i.e., to span the state with respect to some measure μ. This finding is also one of the motivations that motivated us to formulate the problem of searching for the Koopman operator in the measure theoretical framework.

Given the above reasoning, one might still be surprised by the phenomena that linear reconstruction outperforms non-linear reconstruction for the Duffing case (*Otto and Rowley*, 2019a), since there are two attractors but only requires *three* observables. Indeed, for the Duffing problem, with mild assumptions, we explicitly show that one only needs three Koopman observables with *linear reconstruction* due to the inherent symmetry around the origin between the two basins of attraction.

Denoting the two basins of attraction as $\mathcal{S}_{1,2}$, assume we can find a two dimensional (for centered attractor) Koopman invariant subspace on just one of the basins of attraction, say $\mathcal{S}_1$. Note that for convenience, we consider the uncentered attractor for $\mathcal{S}_1$ so it would be a three-dimensional Koopman invariant subspace $\text{span}\{1, \phi_{1,1}, \phi_{1,2}\}$ defined on $\mathcal{S}_1$. Moreover, we also assume such a Koopman invariant subspace *spans* the state variable within $\mathcal{S}_1$, i.e., $\exists \mathbf{w}_{1,1}, \mathbf{w}_{1,2}, \mathbf{b}_1 \in \mathbb{R}^{2\times 1}$ such that $\mathbf{x} = \mathbf{w}_{1,1}\phi_{1,1} +$

$\mathbf{w}_{1,2}\phi_{1,2} + \mathbf{b}_1$.

Then, consider the following extensions with three observables ϕ_1, ϕ_2, ϕ_3 defined on $\mathcal{M}$ where $\mathcal{S}_1 \bigcup \mathcal{S}_2$ is full measure,

$$\phi_1 = \begin{cases} (2\chi_{\mathcal{S}_1} - 1)\phi_{1,1} \circ (2\chi_{\mathcal{S}_1} - 1), & \text{if } \mathbf{x} \in \mathcal{S}_1 \bigcup \mathcal{S}_2, \\ 0, & \text{otherwise.} \end{cases}, \tag{5.34}$$

$$\phi_2 = \begin{cases} (2\chi_{\mathcal{S}_1} - 1)\phi_{1,2} \circ (2\chi_{\mathcal{S}_1} - 1) & \text{if } \mathbf{x} \in \mathcal{S}_1 \bigcup \mathcal{S}_2, \\ 0, & \text{otherwise.} \end{cases}, \tag{5.35}$$

$$\phi_3 = \begin{cases} 2\chi_{\mathcal{S}_1} - 1, & \text{if } \mathbf{x} \in \mathcal{S}_1 \bigcup \mathcal{S}_2, \\ 0, & \text{otherwise.} \end{cases}. \tag{5.36}$$

Our goal is to show that when the observables $\{1, \phi_{1,1}, \phi_{1,2}\}$ are extended from $\mathcal{S}_1$ to $\mathcal{M}$ as ϕ_1, ϕ_2, ϕ_3 in the above, the corresponding subspace is still invariant to $\mathcal{K}_t$ and such a Koopman invariant subspace *spans* the state variable in a *weak* sense, as discussed before. When $\mathbf{x} \in \mathcal{S}_1$, $2\chi_{\mathcal{S}_1}(\mathbf{x}) - 1 = 1$, we have $\phi_1 = \phi_{1,1}$, $\phi_2 = \phi_{1,2}$, $\phi_3 = 1$, of which the subspace is still invariant to $\mathcal{K}_t$ by assumption (since it is restricted on $\mathcal{S}_1$). Thus $\forall t \in \mathbb{R}^+$, $\forall \phi \in \text{span}\{\phi_{1,1}, \phi_{1,2}, 1\}$, we have

$$\mathcal{K}_t|_{\mathcal{S}_1}\phi \in \text{span}\{\phi_{1,1}, \phi_{1,2}, 1\}. \tag{5.37}$$

Moreover, we have by assumption,

$$\mathbf{x} = \mathbf{w}_{1,1}\phi_{1,1} + \mathbf{w}_{1,2}\phi_{1,2} + \mathbf{b}_1 = \mathbf{w}_{1,1}\phi_1 + \mathbf{w}_{1,2}\phi_2 + \phi_3\mathbf{b}_1. \tag{5.38}$$

Now let us consider the case for $\mathbf{x} \in \mathcal{S}_2$. Given eqs. (5.34) to (5.36) we have $\phi_1(\mathbf{x}) = -\phi_{1,1}(-\mathbf{x})$, $\phi_2(\mathbf{x}) = -\phi_{1,2}(-\mathbf{x})$, $\phi_3(\mathbf{x}) = -1$. Also, due to symmetry, we

have $\forall \mathbf{x} \in \mathcal{S}_2$, $-\mathbf{x} \in \mathcal{S}_1$. Then, we have

$$\mathbf{w}_{1,1}\phi_1 + \mathbf{w}_{1,2}\phi_2 + \phi_3\mathbf{b}_1 = -(\mathbf{w}_{1,1}\phi_{1,1}(-\mathbf{x}) + \mathbf{w}_{1,2}\phi_{1,2}(-\mathbf{x}) + \mathbf{b}_1) = -(-\mathbf{x}) = \mathbf{x}. \quad (5.39)$$

Next, consider any $\mathbf{x} \in \mathcal{S}_2$, $\forall \phi \in \text{span}\{\phi_1, \phi_2, \phi_3\}$, we have $\phi = \sum_{i=1}^{3} \alpha_i \phi_i$, $\alpha_i \in \mathbb{R}$,

$$\begin{aligned} \mathcal{K}_t|_{\mathcal{S}_2}\phi(\mathbf{x}) &= \sum_{i=1}^{3} \alpha_i \phi_i(S(\mathbf{x}, t)) = -(\alpha_1\phi_{1,1}(-S(\mathbf{x}, t)) + \alpha_2\phi_{1,2}(-S(\mathbf{x}, t)) + \alpha_3), \quad (5.40) \\ &= -(\alpha_1\phi_{1,1}(S(-\mathbf{x}, t)) + \alpha_2\phi_{1,2}(S(-\mathbf{x}, t)) + \alpha_3), \\ &= -\mathcal{K}_t|_{\mathcal{S}_1}\phi(-\mathbf{x}), \end{aligned}$$

where the property of symmetry with respect to the origin is used: for any initial condition $\mathbf{x} \in \mathcal{S}_1 \bigcup \mathcal{S}_2$, with the corresponding flow map $S(\cdot, t)$ for the Duffing system, one can have $\forall t \in \mathbb{R}^+$, $-S(\mathbf{x}, t) = S(-\mathbf{x}, t)$. From eq. (5.37), $\forall \mathbf{x}' \in \mathcal{S}_1$, for such ϕ, we have $\mathcal{K}_t|_{\mathcal{S}_1}\phi = \alpha_1\phi_{1,1}(S(\mathbf{x}', t)) + \alpha_2\phi_{1,2}(S(\mathbf{x}', t)) + \alpha_3 = \beta_1\phi_{1,1}(\mathbf{x}') + \beta_2\phi_{1,2}(\mathbf{x}') + \beta_3$ for some $\beta_i \in \mathbb{R}$. Thus, take $\mathbf{x}' = -\mathbf{x}$, for any $\mathbf{x} \in \mathcal{S}_2$, we have

$$\begin{aligned} \mathcal{K}_t|_{\mathcal{S}_2}\phi(\mathbf{x}) &= \beta_1(-\phi_{1,1}(-\mathbf{x})) + \beta_2(-\phi_{1,2}(-\mathbf{x})) - \beta_3, \quad (5.41) \\ &= \beta_1\phi_1(\mathbf{x}) + \beta_2\phi_2(\mathbf{x}) + \beta_3\phi_3(\mathbf{x}) \in \text{span}\{-\phi_{1,1}(-\mathbf{x}), -\phi_{1,2}(-\mathbf{x}), -1\}. \end{aligned}$$

Therefore, the subspace spanned by the set of observables in eqs. (5.34) to (5.36) is invariant to $\mathcal{K}_t$ while the case when $\mathbf{x} \notin \mathcal{S}_1 \bigcup \mathcal{S}_2$ is trivially satisfied as well. Last, notice eqs. (5.38) and (5.39) and the fact that $\mathcal{S}_1 \bigcup \mathcal{S}_2$ is full measure, we have

$$\|\mathbf{w}_{1,1}\phi_1 + \mathbf{w}_{1,2}\phi_2 + \mathbf{b}_1\phi_3 - \mathbf{x}\|_{\mathcal{F}} = 0. \quad (5.42)$$

Thus, with mild assumptions, we have shown that one can search for three observables to construct a finite-dimensional Koopman invariant subspace while *spans* the state variable including the bias, in a weak sense. In the specific case of the Duff-

ing oscillator, linear reconstruction exploits the inherent symmetry in the dynamical system and leads to fewer parameters.

However, it must be noted that we do not imply that eqs. (5.34) to (5.36) are the observables learned by a neural network in reality, especially the non–smooth "cliff" function in eq. (5.36). Rather, we surmise that as long as the third observable contains enough information for the decoder to differentiate the two basins of attraction, two attractors can be accommodated.

5.8 Applications: Modal Analysis and ROM

To demonstrate and analyze the approaches presented herein, three numerical examples are pursued for modal analysis and non-intrusive ROMs.

5.8.1 2D fixed point attractor

Consider the two-dimensional non-linear dynamical system (*Lusch et al.*, 2018a) with a fixed point,

$$\dot{x}_1 = \mu x_1,$$
$$\dot{x}_2 = \lambda(x_2 - x_1^2),$$

where $\mu = -0.05$, $\lambda = -1$. For this low dimensional system with known governing equations, we consider the differential form in section 2.5.2.1, with 1600 states as training samples, sampled from $[-0.5, 0.5]$ using the standard Latin-Hypercube-Sampling method (*McKay et al.*, 1979) for both x_1 and x_2. The embedding of SVD-DMD is not employed in this case since it is not very meaningful. The hyperparameters for training are given in table 5.2.

After we obtained the inferred posterior, we consider Monte Carlo sampling described in section 5.6.4 with $N_{mc} = 100$, $M_{mc} = 10$ to approximate the posterior

Table 5.2: Hyperparameters of differential form model for 2D fixed point attractor.

layer structure	optimizer	learning rate	total epoch	batch size
2-8-16-16-8-2-8-16-16-8-2	Adam	1e-4	20000	128

distribution of the Koopman observables and prediction on the dynamical system given an initial condition $\mathbf{x}_0$. The mean Koopman eigenvalues are $\lambda_1 = -0.99656$ and $\lambda_2 = -0.05049$, and the amplitude and phase angle for the mean eigenfunctions are shown in section 5.8.1, which resembles the analytic solution with $\varphi_1 = x_2 - \lambda x_1^2/(\lambda - 2\mu)$, $\varphi_2 = x_1$.

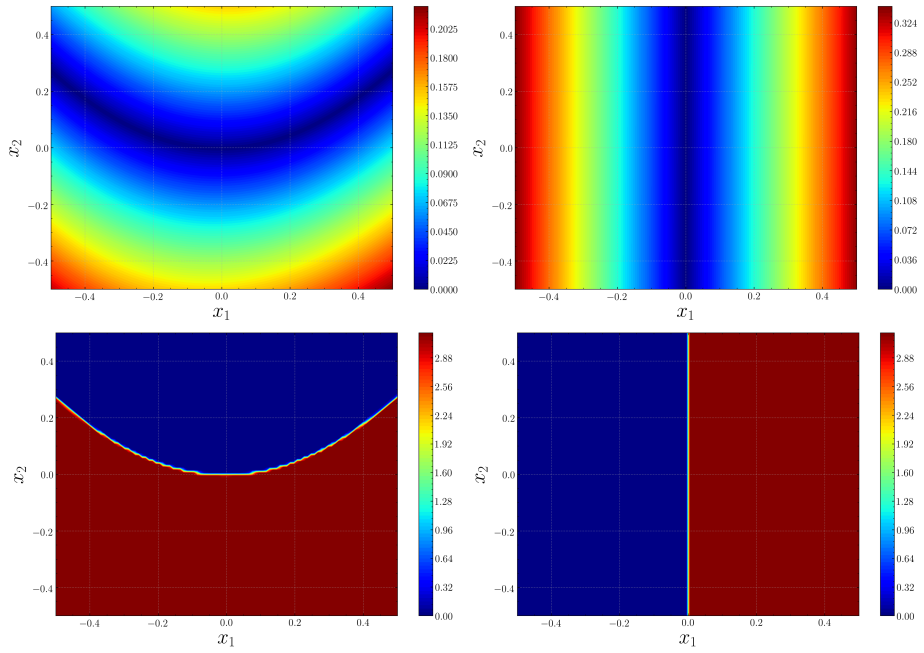


Figure 5.6: 2D fixed point attractor: mean of the variational posterior of learned Koopman eigenfunctions in differential form. Top left: mean of the amplitude of Koopman eigenfunction corresponding to λ_1. Top right: mean of the amplitude of Koopman eigenfunction corresponding to λ_2. Bottom left: mean of the phase angle of Koopman eigenfunction corresponding to λ_1. Bottom right: mean of the phase angle of Koopman eigenfunction corresponding to λ_2.

Finally, since we have a learned a distribution over the Koopman operator, we can obtain the posterior distribution of the predicted dynamics, given an arbitrary unseen

initial condition following section 5.6.4, for example, at $\mathbf{x}_0 = (0.4, -0.4)$. The effect of the number of data samples M on the confidence of the predicted dynamics can be seen in section 5.8.1. Clearly, as more data is collected in the region of interest, the propagated uncertainty of the evolution of the dynamics predicted on the testing data decreases as expected. It is interesting to note that, even when the data is halved, the standard deviation of the Koopman eigenvalue is very small compared to the mean.

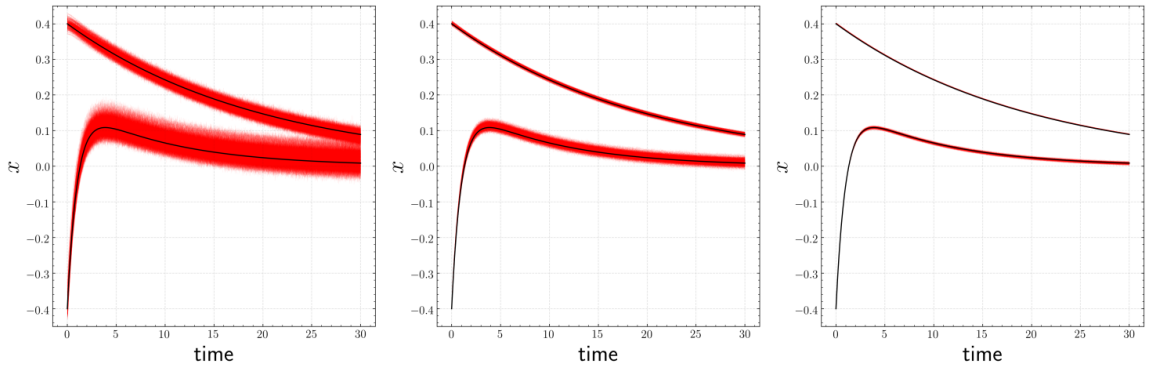


Figure 5.7: 2D fixed point attractor case: Monte Carlo sampling of the predicted trajectory with $\mathbf{x}_0 = [0.4, -0.4]$. Left: 800 (50% of) original data points. Middle: original 1600 data points. Right: 10000 data points.

5.8.2 2D unforced duffing oscillator

Next, we consider the unforced duffing system:

$$\dot{x_1} = x_2,$$
$$\dot{x_2} = -\delta x_2 - x_1(\beta + \alpha x_1^2),$$

where $\delta = 0.5$, $\beta = -1$, $\alpha = 1$. We use 10000 samples of the state $\mathbf{x}$ distributed on $x_1, x_2 \in [-2, 2]$ from Latin-Hypercube-Sampling. We infer the posterior using the differential form model in section 2.5.2.1, with the following hyperparameters in table 5.3.

Table 5.3: Hyperparameters of differential form model for unforced Duffing system.

layer structure	optimizer	learning rate	total epoch	batch size
2-16-16-24-16-16-3-16-16-24-16-16-2	Adam	1e-3	20000	128

To assess the uncertainty in the Koopman eigenfunctions, we draw 100 samples from the variational posterior of $\mathbf{W}_{\mathbf{\Phi}}$ and $\mathbf{K}$. First, the mean of the non-trivial Koopman eigenvalues are $\lambda_{1,2} = -0.535 \pm 0.750i$ and the mean of the third eigenvalue is $\lambda_3 = -2 \times 10^{-5}$. The mean of the module and phase angle of the Koopman eigenfunctions is shown in fig. 5.8. The results are similar to Ref. (*Otto and Rowley*, 2019a) in which a deterministic model is employed. The Koopman eigenfunction associated with λ_3 acts as an indicator of the basin of attraction. Second, to better visualize the effect of uncertainty on unseen data, we normalize the standard deviation of the module of Koopman eigfunctions by the minimal standard deviation over $[-4, 4]$. fig. 5.8 shows a uniformly distributed standard deviation within $[-2, 2] \times [-2, 2]$ where sampling data is distributed, and a drastic increase outside that training region as expected. Note that the area where the normalized standard deviation is larger than ten is cropped.

To obtain the posterior distribution of the evolution of the dynamics predicted by the Koopman operator, we arbitrarily choose an initial condition within the range of training data as $\mathbf{x}(0) = [1.2, 1.2]$. Note that we did not input the model with any trajectory data other than the state and the corresponding time derivative obtained from the governing equation. With Monte Carlo sampling, we obtain the distribution of the trajectory in fig. 5.8. Note that the uncertainty is quite small since it is within the training data region with enough data.

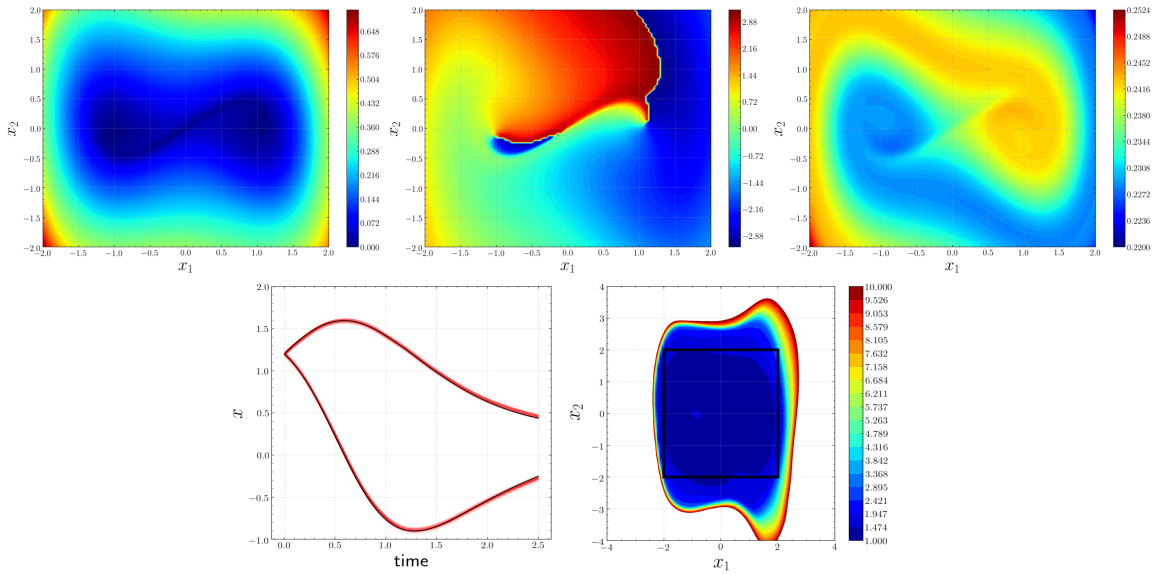


Figure 5.8: 2D unforced Duffing oscillator system case: mean of Koopman observables and predicted trajectory from the differential form model. Top left: mean of the amplitude of Koopman eigenfunction associated with $\lambda_{1,2}$. Top middle: mean of the phase angle of Koopman eigenfunction associated with $\lambda_{1,2}$. Top right: mean of the amplitude of Koopman eigenfunction associated with λ_3. Bottom left: Monte Carlo sampling of the predicted trajectory with initial condition $\mathbf{x}_0 = [1.2, 1.2]$. Bottom right: contour of the normalized standard deviation over $x_1, x_2 \in [-4, 4]$ where the black square represents the boundary of training data.

5.8.3 Transient flow wake behind a cylinder

We consider the velocity field of a two-dimensional laminar flow past a cylinder in a transient regime at a Reynolds number $Re_D = U_\infty D/\nu = 100$, which is above 47, the critical Reynolds number associated with Hopf bifurcation. U_∞ is the freestream velocity, D is the cylinder diameter, and ν is the kinematic viscosity. The transient regime is rather difficult due to the high non-linearity of post-Hopf-bifurcation dynamics between unstable equilibrium and the limit cycle (*Chen et al.*, 2012). Data is generated by solving the two-dimensional incompressible Navier–Stokes equations using OpenFOAM (*Jasak et al.*, 2007). Grid convergence was verified using a sequence of successively refined meshes.

The initial condition is a uniform flow with the freestream velocity superimposed with standard Gaussian random noise and with pressure initialized with Gaussian random noise. Although this initial condition is a rough approximation to the development of true instabilities from equilibrium (*Chen et al.*, 2012), the flow is observed to converge to a quasi-steady solution after a few steps rapidly, and then starts to oscillate and form a long separation bubble with two counter-rotating vortices. The first 50 POD snapshots of velocity are considered with the kinetic energy captured up to 99%. We sample 1245 snapshots of data on the trajectory starting from the unstable equilibrium point to the vortex shedding attractor with $\Delta t = 0.1 t_{ref} = 0.1 D/U_\infty$ where the characteristic advection time scale $t_{ref} = D/U_\infty = 2$ sec. The first 600 snapshots are considered as training data, i.e., $0 \leqslant t \leqslant 60 t_{ref}$.

To further analyze the robustness of the model to noisy data, Gaussian white noise is added [8] to the temporal data of POD coefficients by considering a fixed signal-to-noise ratio as 5%, 10%, 20%, 30%, for each component at each time instance. The model performance is evaluated by predicting the entire trajectory with the (noisy)

[8] Note that noise was not added to the original flow field since taking the dominant POD modes on the flow field would contribute to de-noising.

Table 5.4: Hyperparameters of recurrent form model for flow past cylinder at $Re_D = 100$.

layer structure	optimizer	learning rate	total epoch	batch size	T
50-100-50-20-20-20-50-100-50	Adam	1e-3	20000	64	100

initial condition given, including the remaining 645 snapshots. We consider the recurrent form model described in section 2.5.2.2 together with SVD-DMD described in section 5.4.0.2 with the corresponding hyperparameters given in table 5.4. Note that we consider 20 intrinsic modes, i.e., at most ten distinct frequencies can be captured, which are empirically chosen. Also, the finite-horizon window length corresponding to $T = 100$ is $10t_{ref}$, which is much less than the time required for the system to transit from unstable equilibrium to the attractor.

The continuous-time Koopman eigenvalue distribution of 20 modes for the training data with five different signal-to-noise ratios is shown in fig. 5.9. First, all Koopman eigenvalues are stable according to the stable parameterization of $\mathbf{K}$ in eq. (5.6). Second, when noise is added, the eigenvalues are seen to deviate except the one on the imaginary axis with $\lambda = \pm 0.528j$, which corresponds to the dominant vortex shedding frequency on the limiting cycle with $St = \lambda D/(2\pi U_\infty) = 0.168$. This exercise verifies the robustness of the present approach to Gaussian noise.

Recall that we have a posterior distribution of the predicted trajectory of POD components for with five different noise ratios. Monte Carlo sampling of the posterior distribution is shown for the noisy training data in fig. 5.10. Clearly, uncertainty from the data due to the Gaussian white noise is well characterized by the ensemble of Monte Carlo sampling on the distribution. As a side comparison, we also performed vanilla LSTM-RNN with 20 look-back length (*Hochreiter and Schmidhuber*, 1997) with Keras on clean and the noisiest case with 30% signal-to-noise ratio with exactly the same setup. The structure contains two LSTM layers with 100-dimensional sequential output units, one LSTM layer 100-dimensional units, and one Dropout

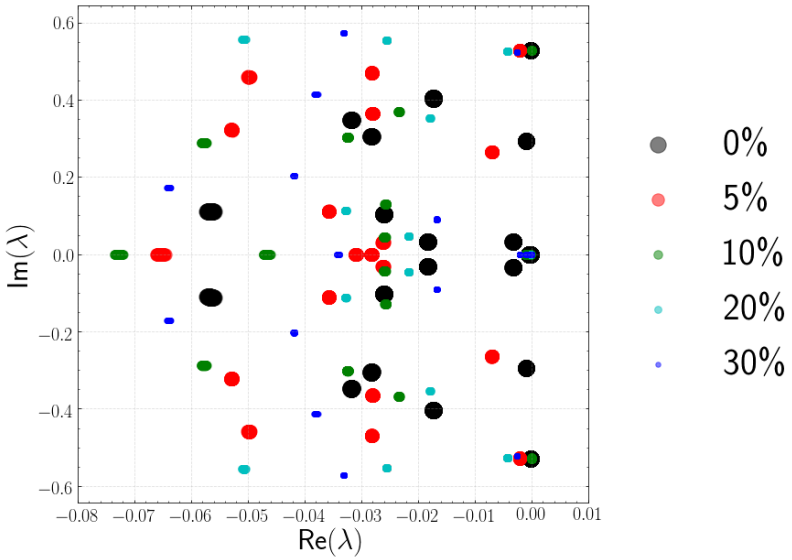


Figure 5.9: Distribution of continuous-time Koopman eigenvalues for fixed signal-to-noise ratio from 5%, 10%, 20%, 30%.

layer (*Srivastava et al.*, 2014) with a linear output layer. We used Adam optimizer with the default setting. We randomly choose 5% data as validation. We stopped batched optimization when it becomes to overfit (i.e., fitting all the noisy data). The results are displayed in the last two rows of fig. 5.10.

In the noiseless case (sixth row in fig. 5.10), we observe that LSTM-RNN can capture the amplitude of the limit cycle as accurate as our model but failed to capture the phase accurately. This might be caused by the lack of multi-step ahead predictions in the loss function of the vanilla LSTM-RNN since the current LSTM-RNN only considers look-back, i.e., history information, with one-step-ahead prediction. Meanwhile, note that even a kernel DMD (one-step state prediction, and no look back) can learn the same dynamics of cylinder flows (noiseless) back in section 4.4.2 very accurately.

In the noisiest case (seventh row in fig. 5.10), it is clear that the current LSTM-RNN model slightly under-predicted the amplitude of the limit cycle and failed to capture the phase of the dynamics significantly. Therefore, we can conclude that

vanilla LSTM-RNN is more sensitive to the noise in the data than our framework.

To further analyze the effect of the difference between the mean of the posterior distribution of the prediction and the ground truth clean trajectory on the flow field, we show the (projected) mean and standard deviation of the predicted POD coefficients at $t = 100t_{ref}$, in fig. 5.11 and fig. 5.12. As seen in fig. 5.11, there is hardly any difference between the mean of the posterior distribution and the ground truth. The contour of standard deviation projected onto the flow field shows a similar pattern to the vortex shedding, and the standard deviation near the wake region is relatively small compared to other domains in the flow field.

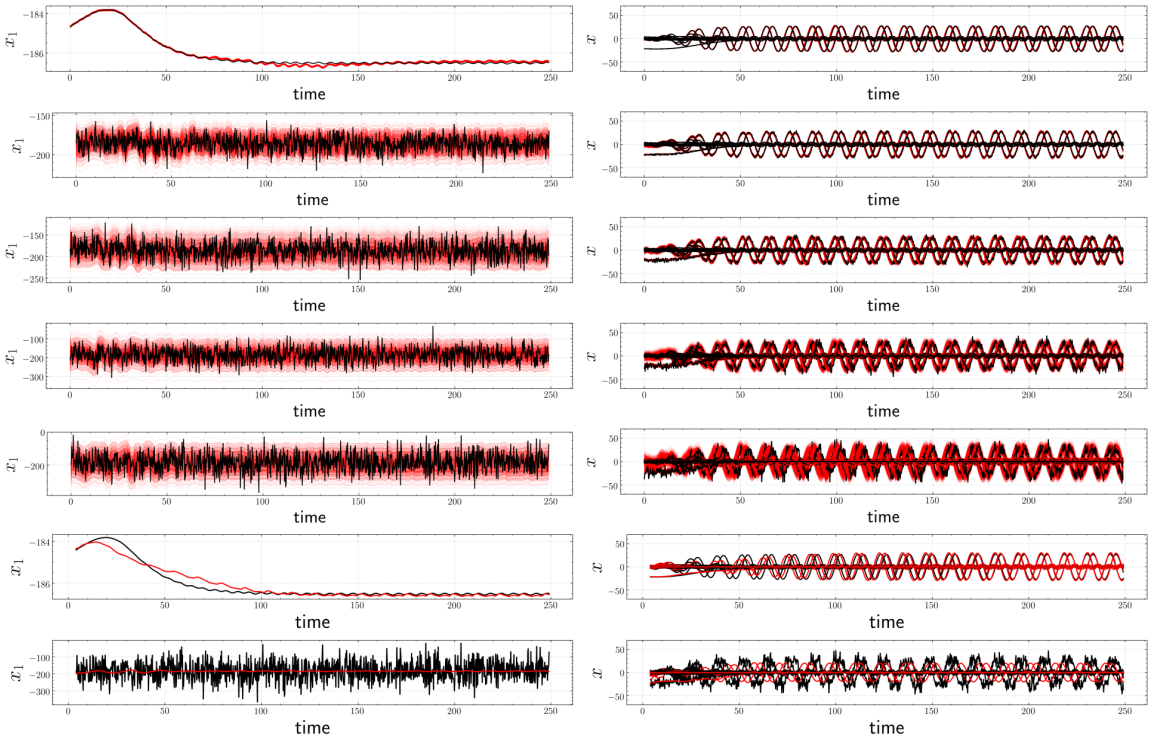


Figure 5.10: Comparison between Monte Carlo sampled distribution of predicted trajectory (Red) and the noisy training data (Black) for signal-to-noise ratio from top to bottom except last two rows as 0% 5%, 10%, 20%, 30%. Bottom two rows: Prediction from vanilla LSTM-RNN with 0% and 30% signal-to-noise ratio, respectively. Left: first POD coefficient. Right: rest 49 POD coefficients.

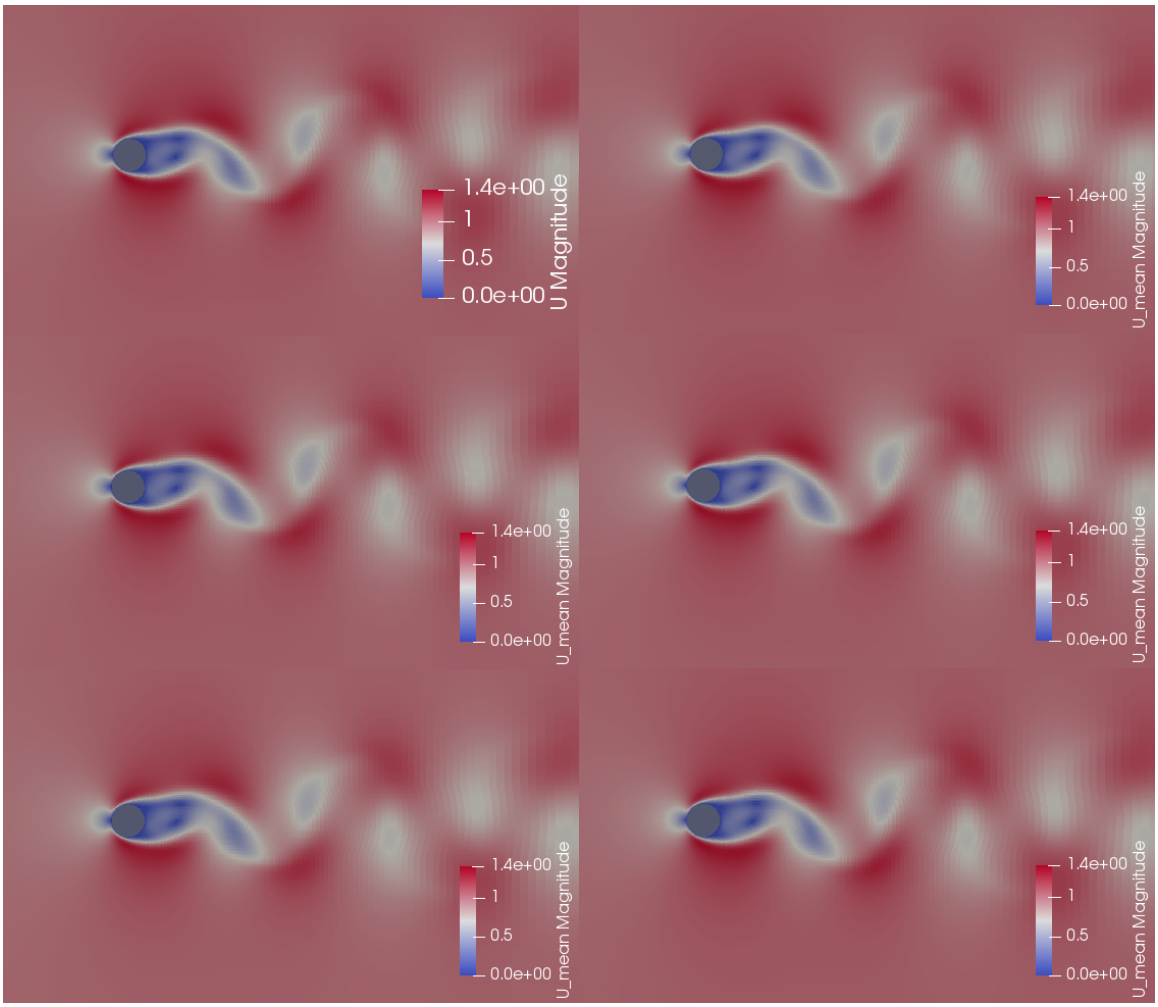


Figure 5.11: Contours of velocity magnitude at $t = 100t_{ref}$. Top left: flowfield projected from the true dominant 50 POD coefficients. Top right: flowfield projected from the 50 POD coefficients of the mean of posterior distribution with clean data. Middle left and right, bottom left and right correspond to noisy training data with signal-to-noise ratio 5%, 10%, 20%, 30%, respectively.

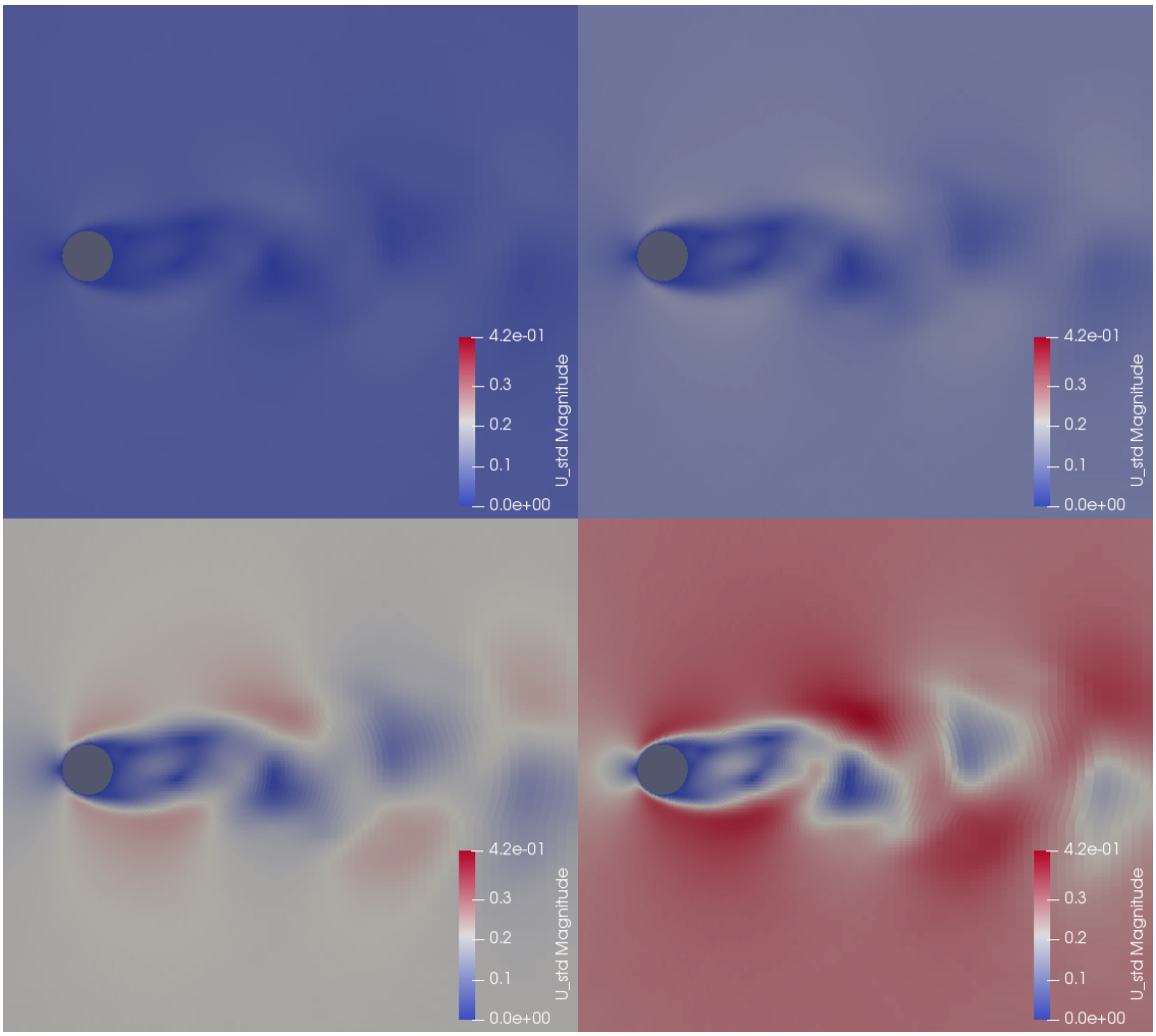


Figure 5.12: Contours of standard deviation of velocity magnitude at $t = 100t_{ref}$. Top left and right, bottom left and right correspond to noisy training data with signal-to-noise ratio 5%, 10%, 20%, 30% respectively.

5.9 Applications: Data-driven Optimal Control for Non-linear System

Given the following general n-dimensional control-affine dynamical system,

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \tag{5.43}$$

where $g(\mathbf{x}) \in \mathbb{R}^{n \times N_I}$, N_I is the number of independent input for the system, we are interested in finding a full-state feedback control law $\mathbf{u}(\mathbf{x})$ such that it minimizes the following cost function, given initial condition $\mathbf{x}_0 \in \mathbb{R}^n$,

$$J(\mathbf{u}, \mathbf{x}_0) = \int_0^\infty \left(\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right) dt \tag{5.44}$$

constrained by the system dynamics in eq. (5.43). For purely linear systems, the optimal solution is provided by the *linear quadratic regulator* (LQR). For a non-linear system, the standard approach (*Khalil and Grizzle*, 2002) starts with a linear approximation of the system around the origin and then follows the LQR.

To leverage the Koopman operator for controlling non-linear systems, we follow Kaiser's idea of Koopman LQR control (*Kaiser et al.*, 2017) by first learning a Koopman decomposition for the autonomous system with our proposed deep learning framework with linear reconstruction in section 5.4. Next, we formulate the optimal control problem in the Koopman representation and apply state-dependent LQR to approximate the optimal feedback control law.

1. Perform simulations of the non-linear system with a certain distribution of initial condition $\mathcal{X}$,

$$\mathbf{x}(0) \triangleq \mathbf{x}_0 \sim \mathcal{X} \tag{5.45}$$

2. Learning a Koopman representation for the autonomous dynamics,

$$\dot{\mathbf{x}} = f(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{5.46}$$

such that by lifting the system into a higher dimensional space with a set of non-linear mapping

$$\mathbf{\Phi} : \mathbb{R}^n \mapsto \mathbb{R}^{N_K} \tag{5.47}$$

we have

$$\dot{\mathbf{\Phi}} = \mathbf{A}\mathbf{\Phi}, \quad \mathbf{x} = \mathbf{B}\mathbf{\Phi} + \overline{\mathbf{x}}, \tag{5.48}$$

where $\mathbf{A} \in \mathbb{R}^{N_K \times N_K}$, $\mathbf{B} \in \mathbb{R}^{n \times N_K}$ and $\overline{\mathbf{x}}$ is a constant vector from the neural network. Once we have the mapping Φ, the corresponding control-affine system can be rewritten as

$$\dot{\mathbf{\Phi}} = \mathbf{A}\mathbf{\Phi} + \nabla_{\mathbf{x}}\mathbf{\Phi} \cdot g(\mathbf{x})\mathbf{u} = \mathbf{A}\mathbf{\Phi} + \mathbf{B}_u(\mathbf{x})\mathbf{u} \tag{5.49}$$

3. Consider the original infinite horizon LQR problem for n-dimensional *non-linear* system,

$$\begin{aligned}
\mathbf{u}^*(t) &= \arg\min_{\mathbf{u}(t)} J(\mathbf{u}, \mathbf{x}_0) = \arg\min_{\mathbf{u}(t)} \int_0^\infty \left(\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right) dt, && (5.50)\\
&= \arg\min_{\mathbf{u}(t)} \int_0^\infty \left((\mathbf{B}\mathbf{\Phi} + \overline{\mathbf{x}})^\top \mathbf{Q}(\mathbf{B}\mathbf{\Phi} + \overline{\mathbf{x}}) + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right) dt, && (5.51)\\
&= \arg\min_{\mathbf{u}(t)} \int_0^\infty \left((\mathbf{B}\mathbf{\Phi} - \mathbf{B}\mathbf{\Phi}(\mathbf{0}))^\top \mathbf{Q}(\mathbf{B}\mathbf{\Phi} - \mathbf{B}\mathbf{\Phi}(\mathbf{0})) + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right) dt, && (5.52)\\
&= \arg\min_{\mathbf{u}(t)} \int_0^\infty \left((\mathbf{\Phi} - \mathbf{\Phi}(\mathbf{0}))^\top \mathbf{Q}_{\mathbf{\Phi}}(\mathbf{\Phi} - \mathbf{\Phi}(\mathbf{0})) + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right) dt, && (5.53)
\end{aligned}$$

which becomes an infinite horizon LQR problem for N_K-dimensional *linear* system with

$$\mathbf{Q}_{\mathbf{\Phi}} = \mathbf{B}^\top \mathbf{Q}\mathbf{B}, \tag{5.54}$$

and a potentially non-linear control-affine input.

4. While eq. (5.49) is not the *standard linear input system* since the input is introduced *non-linearly*. For such optimal control problem, we consider state-dependent LQR algorithm (sub-optimal) which solves $\mathbf{P}$ in the state-dependent Ricatti (SDRE) equation,

$$\mathbf{Q}_{\mathbf{\Phi}} + \mathbf{P}\mathbf{A} + \mathbf{A}^{\top}\mathbf{P} - \mathbf{P}\mathbf{B}_u(\mathbf{x})\mathbf{R}^{-1}\mathbf{B}_u^{\top}(\mathbf{x})\mathbf{P} = \mathbf{0}, \tag{5.55}$$

then the we have the control feedback law as

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{R}^{-1}\mathbf{B}_u^{\top}(\mathbf{x})\mathbf{P}(\mathbf{\Phi}(\mathbf{x}) - \mathbf{\Phi}(\mathbf{0})). \tag{5.56}$$

5.9.1 Stabilization of two dimensional non-linear system

Here we revisit the aforementioned toy example in section 5.8.1. However, here we choose a set of parameters such that the non-linear system without actuation would become *unstable*. To stabilize the non-linear system, we consider augmenting the system with a linear actuation,

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mu x_1 \\ \lambda(x_2 - x_1^2) \end{bmatrix} + \mathbf{B}\mathbf{u}, \tag{5.57}$$

where $\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mu = -0.05$, $\lambda = 1$. Hence, x_1 is naturally stable while x_2 is unstable. The origin is an unstable equilibrium. The optimal control is to stabilize the system on the origin while the energy from the input is still minimized in some sense.

The training data is prepared as 22,500 data pairs: $\{\mathbf{x}_i, \dot{\mathbf{x}}_i\}_{i=1}^{22500}$ which is sampled by Latin hypercube sampler within $[-10, 10]^2$. Note that no time integration on the

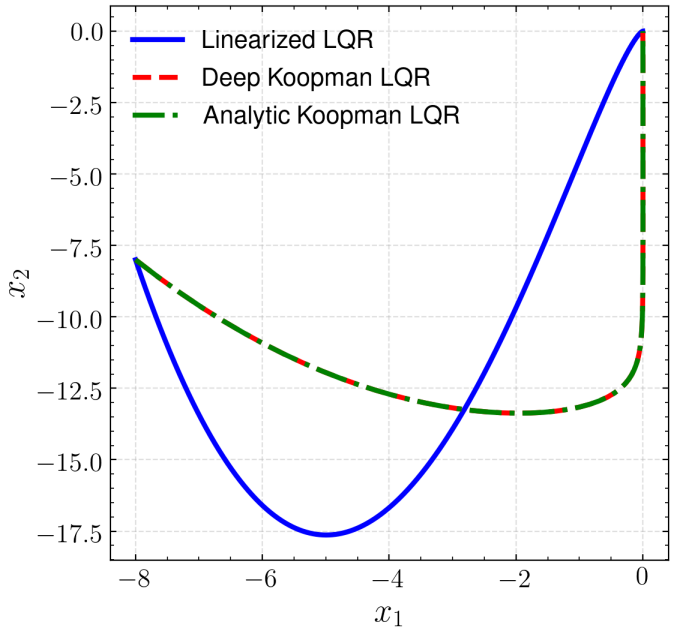


Figure 5.13: Comparison of three controllers with $\mathbf{x}_0 = [-8, 8]$ within training regime: controlled trajectory in x_1 and x_2 phase plane.

system is performed. The layer structure of the feedforward neural network for the encoder $\Phi(\mathbf{x})$ is 2-64-64-64-3 while the decoder is a linear affine mapping. Swish activation function is chosen, and Adam optimizer with learning rate 10^{-3} is used for mini-batched stochastic optimization with batch size as 512.

We initiate the system at $\mathbf{x}_0 = [-8, 8]$ and consider $\mathbf{Q} = \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. In addition, we also perform traditional linearized LQR control on the same problem. As shown in fig. 5.13, Deep Koopman LQR converges to analytic Koopman LQR perfectly. From fig. 5.14, comparing to traditional linearized LQR, Koopman LQR achieves a factor of three reduction in J with 50% faster in the time required for stabilization.

As a sanity check, we take the initial condition $\mathbf{x}_0$ close to the origin so that the traditional linearization becomes a good approximation of the non-linear dynamics. From fig. 5.15, the controlled system starting from $\mathbf{x}_0 = [0.1, 0.1]$ with Koopman LQR

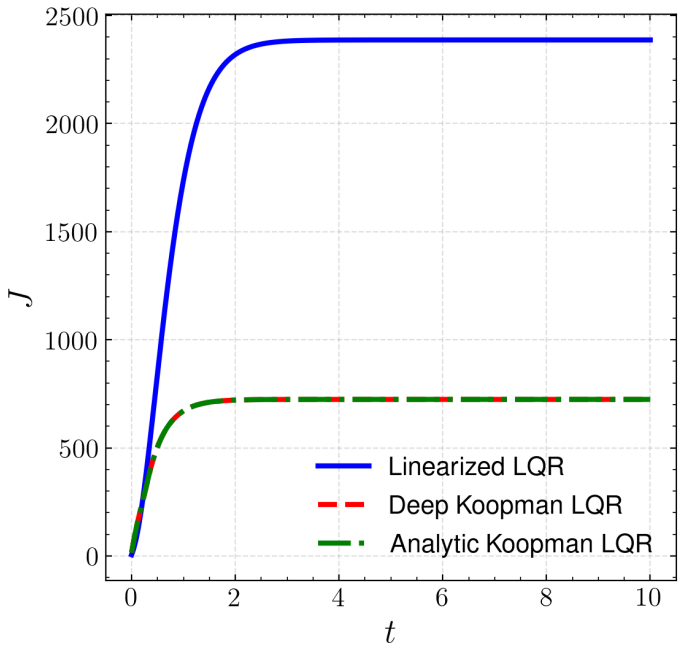


Figure 5.14: Comparison of three controllers with $\mathbf{x}_0 = [-8, 8]$ within training regime: cost functional vs time t

controller is very close to that from the linearized LQR controller. As a result, fig. 5.16 shows that three controllers share almost the same behavior of cost functional.

Next, we consider a more challenging testing problem with initial condition $\mathbf{x}_0 = [-20, 20]$, which starts outside the training regime. As shown in figs. 5.17 and 5.18, a factor of 7 reduction in J is achieved. Moreover, it is interesting to note that the cost functional of Deep Koopman LQR is even slightly smaller than the analytic Koopman LQR for the unseen regime.

Besides, we investigate the distribution of cost depending on the initial state $\mathbf{x}_0$ on the regime $[-20, 20]^2$ which is twice as big as our training regime $[-10, 10]^2$. As shown in fig. 5.19, it is somewhat surprising that Deep Koopman LQR performs *as well as* the analytic Koopman LQR far outside the training regime.

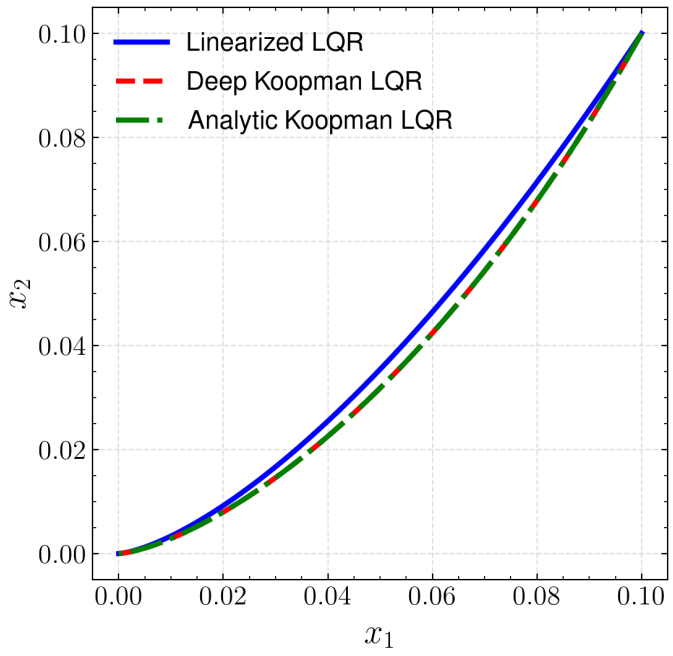


Figure 5.15: Comparison of three controllers with $\mathbf{x}_0 = [-0.1, 0.1]$ within training regime: controlled trajectory in x_1 and x_2 phase plane.

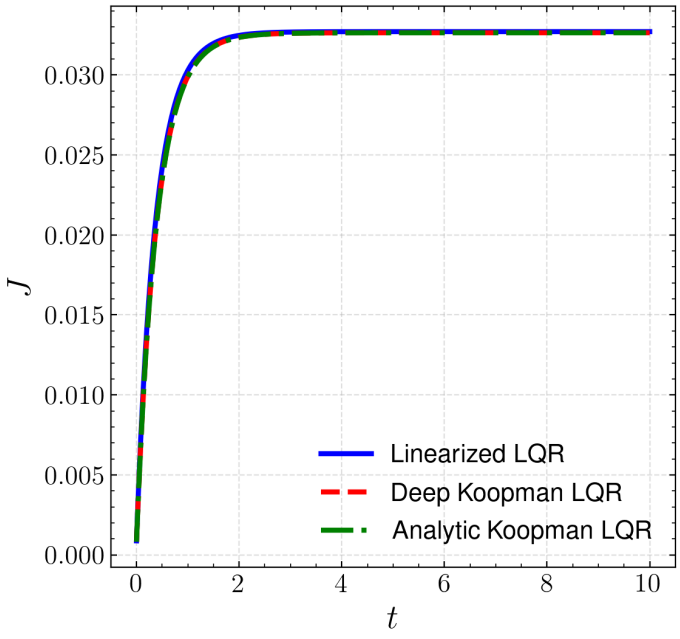


Figure 5.16: Comparison of three controllers with $\mathbf{x}_0 = [-0.1, 0.1]$ within training regime: cost functional vs time t

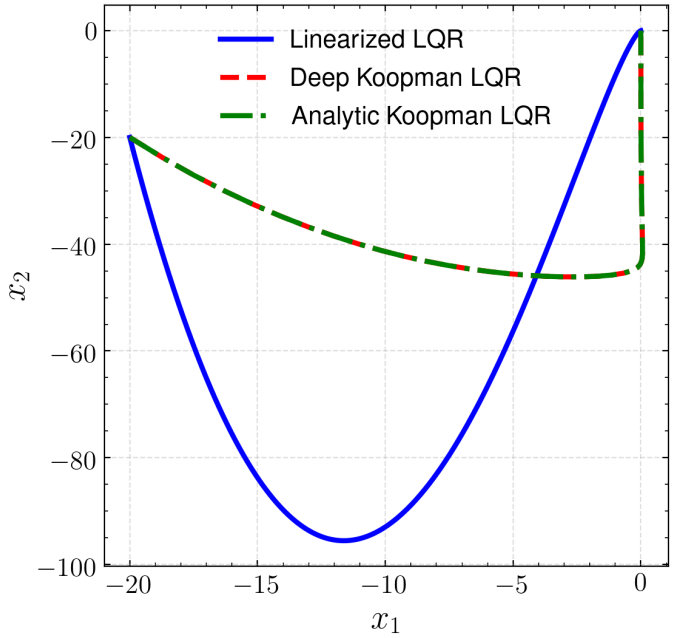


Figure 5.17: Comparison of three controllers with $\mathbf{x}_0 = [-20, 20]$ within training regime: controlled trajectory in x_1 and x_2 phase plane.

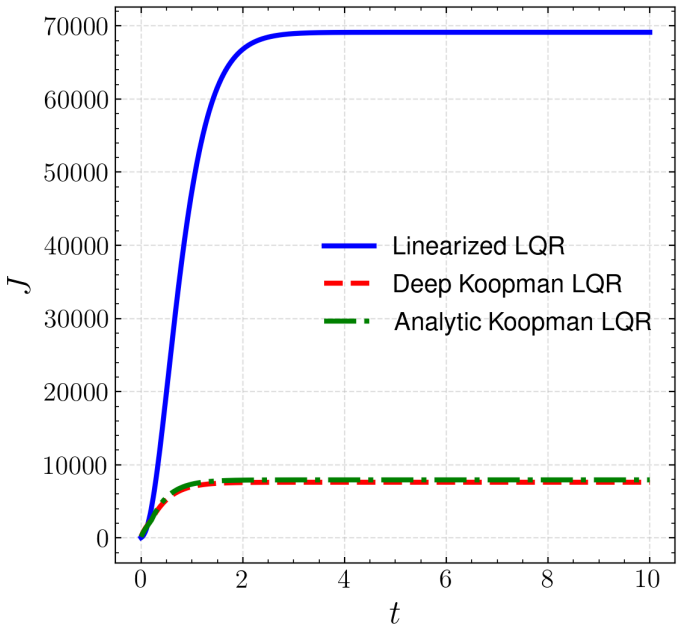


Figure 5.18: Comparison of three controllers with $\mathbf{x}_0 = [-20, 20]$ within training regime: cost functional vs time t

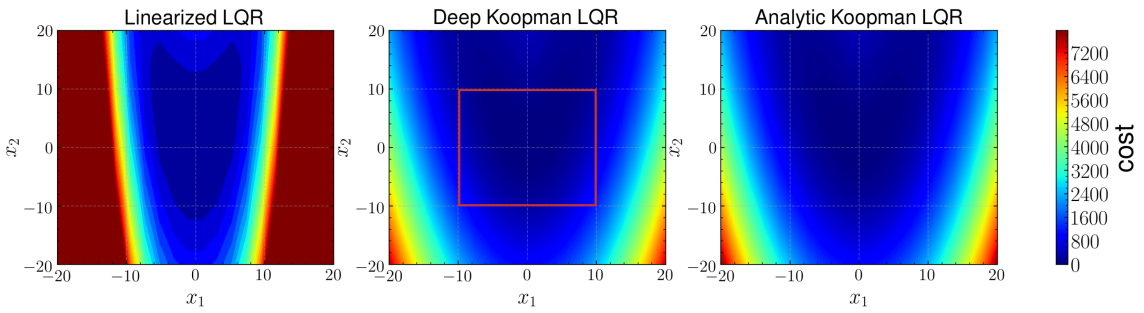


Figure 5.19: Distribution of cost J with respect to $\mathbf{x}_0$ for three different controllers: linearized LQR, Deep Koopman LQR and Analytic Koopman LQR (*Kaiser et al.*, 2017). Red square: boundary of training regime.

5.10 Summary

In this chapter, a probabilistic, stabilized deep learning framework was presented to extract the Koopman decomposition for continuous non-linear dynamical systems. We formulated the deep learning problem from a measure-theoretic perspective with a clear layout of all the assumptions. Two different forms: differential and recurrent, suitable for different situations, were proposed and discussed. A parameterization of the Koopman operator was proposed, with guaranteed stability. Further, a novel deep learning architecture was devised such that the SVD-DMD (*Schmid*, 2010) is naturally embedded. Finally, mean-field variational inference was used to quantify uncertainty in the modeling. For evaluating the posterior distribution, Monte Carlo sampling procedures corresponding to different forms were derived. Finally, the model was evaluated on three continuous non-linear dynamical systems ranging from toy polynomial problems to an unstable wake flow behind a cylinder, from three different aspects: the uncertainty with respect to the density of data in the domain, unseen data, and noise in the data. Meanwhile, vanilla LSTM-RNN is sensitive to the noise in the data and performs worse than our framework. All in all, our results show that the proposed model can capture the uncertainty in all the above cases and is robust to noise.

We also explain the mechanism for learning a Koopman operator with linear reconstruction for a system with multiple fixed-point attractors. We analytically show that one can exploit the inherent symmetry of the dynamical system to reduce the number of Koopman observables.

Lastly, we show an application in data-driven non-linear optimal control on a simple problem. In this problem, the Koopman-based controller is shown to significantly reduce the stabilization time and cost function compared to the linearization-based LQR, and performs as well as the analytically-derived LQR even outside the training region.

CHAPTER VI

Conclusions, Perspectives and Future Work

6.1 Summary and Conclusions

In contrast to the more traditional geometric viewpoint of dynamical systems analysis, Koopman analysis offers an operator-theoretic perspective by describing the dynamics of observables of the system state. The appeal of the Koopman operator lies in the possibility of finding intrinsic coordinates in which the dynamics is globally linear. basin of attraction of the equilibrium point or periodic orbit. This dissertation advances theoretical and computational aspects of the approximation of the Koopman operator using data-driven and physics-informed techniques. Applications are presented in reduced-order modeling, modal analysis, and non-linear optimal control, including problems that involve strong transients and chaotic behavior.

A comprehensive description of the state-of-the-art in data-driven approximation of Koopman operators is presented in chapter II. Motivated by the following three fundamental challenges,

1. Theoretical and quantitative interpretation on the delay structure in the time-delayed dynamic mode decomposition,

2. Interpretable and sparse algorithms for non-linear approximation of Koopman operators (which is critical for the strongly non-linear process),

3. Uncertainty quantification and stability guarantees for Koopman operators for physically realizable non-linear systems,

the major contributions were presented in chapter III, chapter IV, and chapter V, respectively.

Chapter III provides a rigorous theoretical analysis of the structure and numerical conditioning of delay embedding in time-delayed dynamic mode decomposition (TD-DMD) for dynamics on a periodic attractor. We explicitly show that the minimal number of time delays is associated with the sparsity in the Fourier spectrum for scalar dynamical systems. For the vector case, we provide a rank test and a geometric interpretation for the necessary and sufficient conditions for the existence of an accurate linear time-delayed model. Additionally, the explicit exact solution for TD-DMD in the spectral domain is provided for the first time. From a numerical perspective, the effects of the sampling rate and the number of time delays on numerical conditioning are further examined. An upper bound on the condition number is derived, with the implication that conditioning can be improved with additional time delays or decreasing sampling rates. Moreover, it is explicitly shown that the underlying dynamics can be accurately recovered using only a partial period of the attractor, and the underlying reason is identified. Our analysis is first validated in simple periodic and quasi-periodic systems, and sensitivity to noise is also investigated. Finally, issues and practical strategies of choosing time delays in large-scale chaotic systems are discussed and successfully demonstrated on 3D turbulent Rayleigh-Bénard convection. Lastly, we demonstrate the competitive performance of TD-DMD with control (TD-DMDc) on reduced-order modeling of a highly chaotic 2D combustion process against the well-known operator-inference framework. A key benefit of the TD-DMD is that a stable ROM can be guaranteed by removing unstable modes. This is not the case in state of the art techniques such as Operator Inference or LSTM-RNN.

Chapter IV presents a model-agnostic sparsity-promoting framework for extract-

ing an accurate and informative Koopman invariant subspace from data. This framework is based on a) multi-task feature learning to extract the most informative Koopman invariant subspace, and b) pruning spurious Koopman triplets that do not evolve linearly above a certain threshold. We analytically showed an elegant relationship between empirical energy-based criterion, spDMD, and our framework through the lens of non-convex optimization, under the assumption of *using linear observables*, which is the most well-known simplification on Koopman operator of fluid dynamics. As a demonstration, our framework successfully extracts the exact Koopman decomposition on a simple two state non-linear dynamical system. If one is only interested in the post-transient dynamics of the system *on* an attractor, linear observables with time delays are sufficient to extract an informative Koopman-invariant subspace. Thus, the present techniques are evaluated on two unsteady flows which involve strong transients: the widely studied 2D flow over a cylinder at different Reynolds numbers and a 3D turbulent ship air-wake. We studied the parametric dependency of modal shapes with the proposed algorithm on the Reynolds number for the transient cylinder flows. As Re increases from 70 to 130, stable modes become flatter downstream and larger upstream. For the 3D ship air-wake case, two stable modes and one slowly-decaying mode with distinct frequencies and mode shapes are extracted. The visualization clearly shows their dependency on the different parts of the ship geometry. As a comparison against state-of-the-art methods, for the ship air-wake case, we find that spDMD (a widely-used technique in the fluid dynamics community) produces a set of unstable modes, which is a result of the poor approximation of Koopman invariant subspace using linear features. In contrast, our sparsity-promoting framework with KDMD not only isolates the three most dominant coherent structures, but also provides a stable reduced-order model for long-time prediction. At the same time, spDMD blows up immediately beyond the training horizon.

Finally, Chapter V describes a probabilistic deep learning framework and enforce-

ment of stability in approximating the Koopman operator for physically realizable non-linear systems. As we have formalized learning the continuous-time Koopman operator with deep neural networks in a measure-theoretic way, it is natural to induce two types of models: differential and recurrent form, based on how we choose a measure to introduce data, and the availability of the governing equations. We then enforce a structural parameterization that renders the realization of the Koopman operator provably stable, i.e., the most basic measure of physical consistency necessary to model chaotic, noisy systems. Finally, we employ the mean-field variational inference in a hierarchical Bayesian setting for the deep neural nets to quantify uncertainties in the Koopman decomposition and prediction of the dynamics of observables. The effectiveness of our framework is successfully demonstrated on several non-linear dynamical systems, including the Duffing oscillator and an unstable cylinder wake flow with noisy measurements. Our framework shows superior performance over the well-known LSTM-RNN framework. As a side note, we explicitly show that linear reconstruction can be useful for non-linear systems with symmetrical attractors. Lastly, we demonstrated a simple application in non-linear optimal control that leverages the Koopman operators learned by our framework. In this problem, the Koopman-based controller was shown to significantly reduce the stabilization time and cost function compared to the linearization-based LQR, and performs as well as the analytically-derived LQR even outside the training region.

6.2 Perspectives

In contrast to non-linear data-driven modeling approaches (e.g. LSTM-RNN (*Mohan et al.*, 2019), Neural-ODE (*Maulik et al.*, 2020), OpInf (*Peherstorfer and Willcox*, 2016), CNN (*Xu and Duraisamy*, 2019), GAN (*Lee and You*, 2019)), Koopman operator theory brings interpretability and structure that enables flexible imposition of physics information and non-linear optimal control. Generic non-linear data-driven

modeling frameworks are often criticized for being a black-box. Consider a scenario in which one already has a bunch of datasets with varying parameters. Traditional data-driven modeling frameworks target *the task of prediction*, which amounts to an *interpolation within parameter space*. The Koopman operator framework - on the other hand - has the potential to offer *physics-consistent predictions*, *control* and *modal analysis*. The most important contribution of this dissertation is to bring *robustness* and *interpretability* in Koopman operator modeling, which further differentiates it from other non-linear frameworks.

This thesis developed theoretical results for the structure of time delay embedding for linear models on the periodic/quasi-periodic attractors. However, these results are not strictly valid for complex chaotic systems where an infinite number of time delays may be required for a *perfect* prediction. Nevertheless, the theory developed for improvement on numerical conditioning with over-delays is still valid for stabilizing model performance for chaotic systems, as we have demonstrated in the 3D thermal convection problem. It is noted that we only leverage the insights of "over-delays" from our developed theory for modeling complex chaotic systems, when searching for the optimal time delays for a stable model empirically with validation split as shown in 2D combustion process.

As mentioned in the introduction, this thesis is restricted to discrete spectra in approximating the Koopman operator. Therefore, rigorously speaking, compared to approaches such as the SPOD (*Towne et al.*, 2018), and resolvent operators (*McKeon and Sharma*, 2010), the methodology developed in this dissertation may not be the *most* appropriate tool for modal analysis for highly chaotic flows. For systems with dominant discrete spectra, our sparsity-promoting framework is suitable to extract a set of interpretable spectra from the data as demonstrated in the 3D ship air-wake. In terms of more general predictive modeling, the methodology developed in this dissertation can be used as an efficient model with competitive predictive accuracy

as demonstrated by TD-DMDc in the 2D chaotic combustion case.

Since the methodology developed in this dissertation is mainly - though not exclusively - *data-driven*, it is particularly useful in modeling increasingly complex systems for which first-principle models are difficult to obtain or too costly to be practical. This can be very helpful in several domains of application such as flow control, soft robotics, robotic fish, etc. In terms of temporal stability, which has been a notorious problem in projection-based ROM, the methodology developed in this thesis provides the *only* known *robust* framework that guarantees temporal stability. Moreover, with additional modification, one can similarly impose constraints such that Koopman eigenvalues lie on the unit circle. This naturally introduces physics-based constraints for Hamiltonian systems.

6.3 Suggestions for Future Work

Given the above developments in the data-driven approximation of the Koopman operator, we envision several future directions for further applications on non-linear dynamical systems ranging from turbulent flows to robotics:

1. **Data-driven Koopman-based optimal control for robotics**. Recently (*Mamakoukas et al.*, 2019; *Abraham et al.*, 2017; *Kaiser et al.*, 2017) the Koopman operator has been used in data-driven control, especially in scenarios in which accurate governing equations are challenging to obtain via first principles. For such systems, one could perform model-free deep reinforcement learning (DRL) but at the cost of higher sample complexity and no guarantee for control performance even though input constraints will also be difficult to impose. To the contrary, when considering model predictive control (MPC) or linear quadratic regulator (LQR), Koopman-based models can leverage linear, convex optimization to ensure real-time efficiency and global optimality. It has recently been

shown that such an approach leads to a significant reduction in training samples when compared against standard DRL technique (*Hao and Han*, 2020).

2. **Non-linear Koopman analysis for extracting coherent structures in turbulent Flows**. Given the abundant applications of DMD in analyzing and sensing in turbulent flows *Sayadi et al.* (2012); *Alessandri et al.* (2019); *Kramer et al.* (2017), one can develop upgrades with non-linear sparsity-promoting KDMD/EDMD method for improving accuracy and predictability. Since our framework involves mode-by-mode error analysis, Koopman modes will be more linearly consistent than all previous methods. This can lead to a more accurate and interpretable decomposition of fluid flows. As for the fully chaotic, incoherent part of the flows, further research can leverage spectral estimation in signal processing to extract the continuous spectrum (*Arbabi and Mezić*, 2017).

3. **Generalizability of Koopman decomposition in high-dimensional systems.** Most existing Koopman decompositions for (nominal) high-dimensional systems are based on only a *single* trajectory. This may be a good approximation only in ergodic systems. Indeed, the dimensionality of the high dimensional system is only significantly reduced when it falls on an attractor, which can be easily achieved if the simulation is performed long enough. In general, since we do not have access to trajectory data for an arbitrary initial condition, one cannot provide guarantees on the Koopman ROM for a completely unseen initial condition, especially in the initial phase. Thus, it is interesting to analyze the behavior, especially in the transition towards the attractor. If all of the Koopman modes are stable, and if the new initial condition still converges to the same attractor as before, the long time prediction should fall approximately on the same modal shape of the attractor. However, small errors in the amplitude and phase can complicate matters.

APPENDICES

APPENDIX A

Variational Inference

The Bayes rule (*Friedman et al.*, 2001) to update the probability distribution of a parameter θ given data $\mathcal{D}$ is given by

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}, \tag{A.1}$$

where

- $p(\mathcal{D}|\theta)$ represents the likelihood of observing the data $\mathcal{D}$ given the model parameter.
- $p(\theta)$ is called the *prior* distribution. This represents our knowledge of the model parameter θ before observing the data.
- $p(\theta|\mathcal{D})$ is called *posterior* distribution. It represents the updated distribution of parameter θ compared to the prior distribution.
- $p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta$ is called the *evidence*. It represents the marginal probability of the data, and numerical integration for this term is in the θ space, which can be high dimensional.

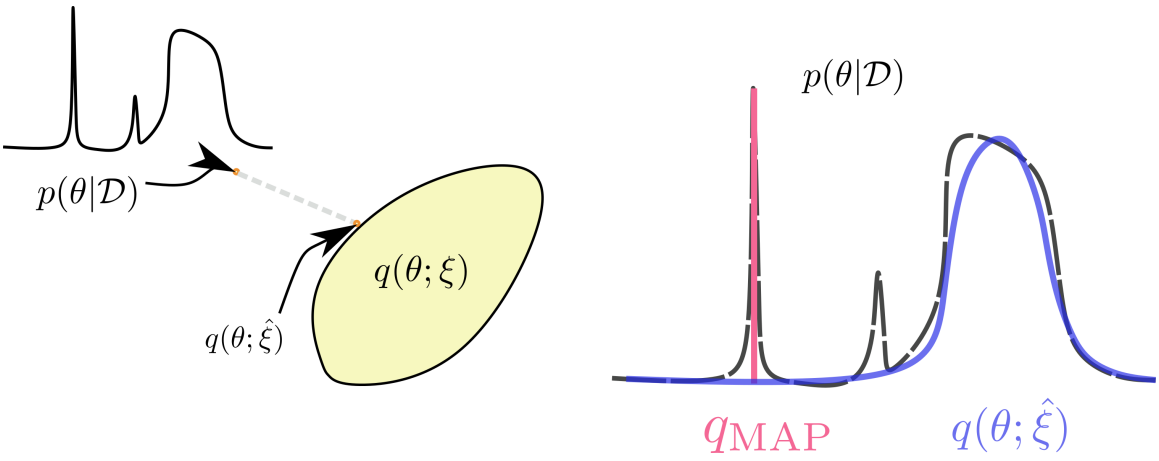


Figure A.1: Left: sketch of variational inference. $p(\theta|\mathcal{D})$ is the posterior. $q(\theta;\xi)$ is a family of probabilistic distributions. $q(\theta;\hat{\xi})$ is the optimal approximation within such family that parameterized by ξ. Right: VI vs MAP approximation where q_{MAP} only finds the maximal location for posterior.

A.1 Introduction to Variational Inference

Instead of explicitly evaluating the posterior $p(\theta|\mathcal{D})$ or using the Monte carlo method to approximate the posterior, variational inference (VI) (*Blei et al.*, 2006) searches for the *best approximation* to the posterior within a predefined function space. An illustration of the main idea compared with the simple maximum a posterior (MAP) approximation is shown in fig. A.1. In the following, we will briefly go through the main idea used in VI for measuring the difference between distributions called *KL divergence* and the practical general computational framework called automatic differentiation variational inference (ADVI) (*Kucukelbir et al.*, 2017).

A.1.1 KL divergence

Originated from information theory, KL divergence measures the difference between two probability distributions p and q, according to the difference between

cross-entropy and its entropy

$$\mathrm{KL}(p||q) \triangleq \int p(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta \tag{A.2}$$

$$= -\int p(\theta) \log q(\theta) dx + \int p(\theta) \log p(\theta) d\theta, \tag{A.3}$$

$$= H(p, q) - H(p), \tag{A.4}$$

where

$$H(p) = -\int p(\theta) \log p(\theta) d\theta = -\mathbb{E}_p[\log(p(\theta))] \tag{A.5}$$

is the Shannon entropy on distribution p. One can simply show KL divergence is non-negative with Jensen's inequality. Intuitively, if $p = q$, then according to eq. (A.2), the KL divergence will be zero. However, it is not a formal distance since it is not symmetric.

An interesting fact that might help the readers on connecting KL divergence with the usual distance metric is the following *Pinsker's inequality*:

$$\delta(p, q) \leqslant \sqrt{\frac{1}{2}\mathrm{KL}(p||q)} \tag{A.6}$$

where the total variation distance is defined as,

$$\delta(p, q) = \sup\{|p(A) - q(A)||A \in \Sigma \text{ is a measurable event}\}, \tag{A.7}$$

It shows that KL divergence relates to an upper bound for the maximal possible difference between two distributions. Thus, *if we can minimize the KL divergence, the two corresponding distributions should be closer to each other.*

A.1.2 Evidence lower bound

Now let us get back to *searching for the best approximation of posterior distribution* $p(\theta|\mathcal{D})$ *within a family of probabilistic distributions*, i.e., $q(\theta;\xi)$. Then, it becomes natural to directly minimize the KL divergence between the parameterized family of distributions as,

$$\hat{\xi} = \arg\min_{\xi} \mathrm{KL}(q(\theta;\xi)||p(\theta|\mathcal{D})) \arg\min_{\xi} = \int q(\theta;\xi) \log \frac{q(\theta;\xi)}{p(\theta|\mathcal{D})} d\theta. \tag{A.8}$$

Unfortunately, naively evaluating eq. (A.8) requires access to the posterior distribution, which is exactly we are looking for. To resolve this issue, we alternatively consider maximizing the so-called *evidence lower bound (ELBO)* or sometimes referred to as *variational lower bound* which shares the relation to KL divergence between our candidate distribution $q(\theta;\xi)$ and ground true posterior distribution $p(\theta|\mathcal{D})$. Recall the KL divergence in eq. (A.8),

$$\begin{aligned}
\mathrm{KL}[q(\theta;\xi)\|p(\theta|\mathcal{D})] &= \int q(\theta;\xi) \log \frac{q(\theta;\xi)}{p(\theta|\mathcal{D})} d\theta \\
&= -\int q(\theta;\xi) \log \frac{p(\theta|\mathcal{D})}{q(\theta;\xi)} d\theta \\
&= -\int q(\theta;\xi) \log \frac{p(\theta,\mathcal{D})}{q(\theta;\xi)p(\mathcal{D})} d\theta \\
&= -\left(\int q(\theta;\xi) \log \frac{p(\theta,\mathcal{D})}{q(\theta)} d\theta - \int q(\theta;\xi) \log p(\mathcal{D}) d\theta \right) \\
&= -\int \left(q(\theta;\xi) \log \frac{p(\theta,\mathcal{D})}{q(\theta;\xi)} d\theta + \log p(\mathcal{D}) \int q(\theta;\xi) \right) d\theta \\
&= -\int q(\theta;\xi) \log \frac{p(\theta,\mathcal{D})}{q(\theta;\xi)} d\theta + \log p(\mathcal{D}) \\
&= -\mathrm{ELOB} + \log p(\mathcal{D})
\end{aligned}$$

where ELBO can be rewritten as,

$$\text{ELBO} = \mathbb{E}_{q(\theta;\xi)}[\log p(\mathcal{D}|\theta)] - \text{KL}[q(\theta;\xi)\|p(\theta)] \tag{A.9}$$

as the *variational lower bound* or *evidence lower bound*. The first term is related to the likelihood while the second term is a regularization term that measures the difference between variational posterior and prior belief. Note that $p(\mathcal{D})$ is a constant (unknown), so only variational lower bound is related to the KL divergence. Moreover, now we *have access to every term inside the ELBO* comparing to the previous KL divergence in eq. (A.8). Therefore we choose to vary ξ in $q(\theta;\xi)$ to maximize ELBO in order to minimize the KL divergence.

A.2 Automatic Differentiation Variational Inference

In this thesis, we employ a computational framework of variational inference called *automatic differentiation variational inference* (*Kucukelbir et al.*, 2017) that enables an automatic variational inference in Tensorflow. It is designed to be general and flexible for arbitrarily large neural networks.

In general, ADVI performs these steps:

1. automatically transforms the latent variable θ to the real coordinate space $\mathbb{R}^p$ as $\hat{\theta}$, where p is the dimension of the parameter space.

2. computes the ELBO with Monte Carlo integration which only requires sampling from the variation distribution $q(\hat{\theta};\xi)$.

3. apply stochastic gradient ascent to maximize the ELBO.

In the following, we will briefly provide a collection of the tricks and ideas of ADVI. For more specific details, we refer to *Kucukelbir et al.* (2017).

A.2.1 Transforming θ to $\hat{\theta}$

Consider a transformation that removes the constraints on θ,

$$+\ \hat{\theta} = T(\theta) \tag{A.10}$$

the corresponding joint density function of data $\mathcal{D}$ and transformed parameter $\hat{\theta}$ would be,

$$p(\hat{\theta}, \mathcal{D}) = p(T^{-1}(\hat{\theta}), \mathcal{D}) \left| \det \frac{dT^{-1}}{d\hat{\theta}}(\hat{\theta}) \right|, \tag{A.11}$$

$$= p(\theta, \mathcal{D}) |\det J_{T^{-1}}(\hat{\theta})| \tag{A.12}$$

where the last term on the above right hand side associated with the Jacobian is to make sure the volume property is preserved after the transformation. Similar relationship also holds for their corresponding variational posterior.

A.2.2 Variational approximation in $\hat{\theta}$ space

As a natural choice, we parameterize the family of distribution $q(\hat{\theta}; \xi)$ with *multivariate normal* distribution (MVN). It should be noted that by considering Gaussian in the transformed space $\hat{\theta}$, implicitly, non-Gaussian variational distribution is in the original latent space θ. Here are some common choices of covariance in the MVN:

1. *Mean-field Gaussian.* A simple assumption is to consider all variables are statistically independent, i.e., $q(\hat{\theta}; \xi)$ has a diagonal covariance matrix:

$$q(\hat{\theta}; \xi) = \mathcal{N}(\hat{\theta} | \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)) = \prod_{k=1}^{p} \mathcal{N}(\hat{\theta}_k | \mu_k, \sigma_k^2) \tag{A.13}$$

   However, note that the variance should always be positive. So we choose the transformation $\omega = \log(\sigma)$ such that *all optimization variables becomes un-*

constrained, i.e., $\xi = \begin{pmatrix} \mu_1 & \dots & \mu_p, \omega_1 & \dots & \omega_p \end{pmatrix} \in \mathbb{R}^{2p}$. That is to say, the p-dimensional random distribution is uniquely described by a deterministic $2p$-dimensional variational parameters.

2. *Full-rank Gaussian.* Note that, removing the correlation among θ could be potentially harmful to the accuracy of model. So a better option is to posit a full rank Gaussian variational approximation with $\xi = (\boldsymbol{\mu}, L)$, where L is the Choleksy factorization of Σ. Note that L has $p(p+1)/2$ components. Therefore, together with $\boldsymbol{\mu}$, we have $p + p(p+1)/2$ parameters for ξ. Unfortunately, with full-rank approximation, the number of variational parameters becomes $O(p^2)$, instead of $O(p)$ that comes from the previous mean-field approximation. Therefore, full-rank approximation is seldom considered for large deep neural networks.

A.2.3 ELBO in $\hat{\theta}$ space

ELBO in the transformed space is the following:

$$\begin{aligned}
\text{ELBO}(\xi, \mathcal{D}) &= \int q(\theta; \xi) \log \left[\frac{p(\mathcal{D}, \theta)}{q(\theta; \xi)} \right] d\theta && \text{(A.14)} \\
&= \int q(\hat{\theta}; \xi) \log \left[\frac{p(\boldsymbol{x}, T^{-1}(\hat{\theta})) \left| \det \frac{dT^{-1}(\hat{\theta})}{d\hat{\theta}} \right|}{q(\hat{\theta}; \xi)} \right] d\hat{\theta} && \text{(A.15)} \\
&= \mathbb{E}_{q(\hat{\theta}; \xi)} \left[\log p(\boldsymbol{x}, T^{-1}(\hat{\theta})) + \log \left| \det J_{T^{-1}}(\hat{\theta}) \right| \right] + \mathbb{H}(q(\hat{\theta}; \xi)), && \text{(A.16)}
\end{aligned}$$

where $\mathbb{H}(q(\hat{\theta}; \xi)) = -\int q(\hat{\theta}; \xi) \log q(\hat{\theta}; \xi) d\hat{\theta}$ is the *entropy* in information theory.

A.2.4 Reparameterization trick

Even though we made a simplification on the variational posterior being MVN, eq. (A.16) still contains an expectation with respect to the *unknown* variational posterior distribution. Therefore, it is often difficult to find the closed-form expression.

However, we know how to sample from the standard normal distribution. Moreover, since we assumed MVN, there is an analytical transformation S_ξ that "regularize" the randomness of the variational parameters $\hat{\theta}$ into the standard normal distribution. Therefore, we can convert the *unknown* variational posterior (which is assumed to be MVN) into a standard Gaussian, i.e., zero mean and unit standard deviation.

- In the mean-field case, $\eta = S_\xi(\hat{\theta}) = \text{diag}(\exp(\boldsymbol{\omega}))^{-1}(\hat{\theta} - \boldsymbol{\mu})$
- In the full-rank case, the standardization is $\eta = S_\xi(\hat{\theta}) = \mathbf{L}^{-1}(\hat{\theta} - \boldsymbol{\mu})$,

where $\mathbf{L}$ is the Cholesky factorization matrix of the covariance matrix in the multivariate Gaussian distribution.

Now we rewrite eq. (A.16) with η so that

$$q(S_\xi^{-1}(\eta)) \left|\det J_{S_\xi^{-1}}(\eta)\right| = q(\eta; \xi) = \mathcal{N}(\eta|\mathbf{0}, \mathrm{I}). \tag{A.17}$$

Thus, we can minimize the following expectation,

$$\xi^* = \arg\max_\xi \mathbb{E}_{\mathcal{N}(\eta;\mathbf{0},\mathrm{I})} \left[\log p(\boldsymbol{x}, T^{-1}(S_\xi^{-1}(\eta))) + \log \left|\det J_{T^{-1}}(S_\xi^{-1}(\eta))\right|\right] + \mathbb{H}(q(\hat{\theta}; \xi)). \tag{A.18}$$

Notice that the expectation is computed over the standard Gaussian, which is independent of the ξ. Therefore, we can take the ∇ inside the expectation. Recall that we have $\xi = (\boldsymbol{\mu}, \boldsymbol{\omega})$ for mean-field approximation or $\xi = (\boldsymbol{\mu}, \mathbf{L})$ for full rank approximation. Thus, we can derive the gradients for ξ analytically: with respect to $\boldsymbol{\mu}$,

$$\nabla_\mu \mathcal{L} = \mathbb{E}_{\mathcal{N}(\eta)} \left[\nabla_\theta \log p(x, \theta) \nabla_{\hat{\theta}} T^{-1}(\hat{\theta}) + \nabla_{\hat{\theta}} \left|\det J_{T^{-1}}(\hat{\theta})\right|\right]. \tag{A.19}$$

and the gradients with respect to the covariance for mean-field Gaussian:

$$\nabla_\omega \mathcal{L} = \mathbb{E}_{\mathcal{N}(\eta)} \left[(\nabla_\theta \log p(x, \theta) \nabla_{\hat{\theta}} T^{-1}(\hat{\theta}) + \nabla_{\hat{\theta}} \left|\det J_{T^{-1}}(\hat{\theta})\right|)\eta^\top\right] + (\mathbf{L}^{-1})^\top, \tag{A.20}$$

and that for full-rank Gaussian:

$$\nabla_\omega \mathcal{L} = \mathbb{E}_{\mathcal{N}(\eta)} \left[(\nabla_\theta \log p(x,\theta) \nabla_{\hat{\theta}} T^{-1}(\hat{\theta}) + \nabla_{\hat{\theta}} \left| \det J_{T^{-1}}(\hat{\theta}) \right|) \eta^\top \mathrm{diag}(\exp(\omega)) \right] + \mathbf{1}. \tag{A.21}$$

Finally, to approximate the expectation, one draws samples from the *standard Gaussian* and evaluates the empirical mean of the gradients within the expectation. Such approximation of the gradient is unbiased. We suggest considering only one sample for speed. However, it should be noted that increasing the number of samples would decrease the variances in the computed gradients, with the scaling rate as $O(1/\sqrt{S})$. S is the number of samples in the Monte Carlo. To update the gradients, one can use Adam or standard SGD algorithm.

A.2.5 Computational complexity

The above ADVI framework has a computational complexity as $O(NSK)$. N is the number of data points. S is the number of Monte Carlo samples, K is the number of variational parameters in the model. It is important to note that we choose mean-field approximation due to efficiency for large models, e.g., deep neural networks. Note that the computational cost is reduced to $O(BSK)$ for the mini-batched version, where B is the batch size.

APPENDIX B

Deep Learning and Bayesian Neural Networks

B.1 Deep Learning

Deep learning is a research area within the machine learning community, which aims to learn meaningful relationship from data using deep neural networks (DNN). It has been an active community in computer science in the recent two decades, especially with the ever growing computational power and improved stochastic optimization techniques that enable the possibility of efficient training on huge datasets. The core of deep learning is *representation learning*, i.e., learning a set of features that encapsulate information in the data that determine the input-output relationship. Notably, a huge amount of success has been reported in the subject of object detection and image recognition. For example, deep convolutional neural network (D-CNN) that embeds the translation invariance is reported to surpass the human expert's performance in the task of image classification (*Krizhevsky et al.*, 2017) .

In the following sections, several basic engines behind deep learning will be briefly introduced.

B.1.1 Feedforward Neural Network

Feedforward neural network (FNN) is the most fundamental deep learning architecture. Typically, one can regard any feedforward neural networks with more than five layers as *deep neural nets*. We define a L-layer feedforward neural network as a vector-valued function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ with $L-1$ hidden layers and last output layer as linear affine mapping. Such L-layer FNN is compositional. Each hidden layer is defined by following recursive expression:

$$\eta^l = \sigma_l(W_l \eta^{l-1} + b_l), \quad l = 1, \ldots, L-1, \tag{B.1}$$

where η^0 stands for the input of the neural network, i.e., $\eta^0 = x$, $\eta^l \in \mathbb{R}^{n_l}$. $n_l \in \mathbb{N}^+$ is the number of hidden units in layer l. σ_l is the activation function of layer l. The last output layer is free of any activation function, since it might introduce unwanted constraints on the output. Therefore, the last output layer is defined as a linear affine mapping:

$$f(x; W^L, b^L) = \eta^L = W_L \eta^{L-1} + b_L, \tag{B.2}$$

where parameters of the neural network are $W^L = \{W_i\}_{i=1,\ldots,L}$, $b^L = \{b_i\}_{i=1,\ldots,L}$.

To give an example, let us consider using two hidden layers, i.e., $L = 3$, and keep the number of units the same for all hidden layers. Then the complete expression for the neural network model from the input x to output $\hat{y}$ is,

$$\hat{y} = f(x; W, b) = W_3 \sigma(W_2 \sigma(W_1 x + b_1) + b_2) + b_3, \tag{B.3}$$

where $x \in \mathbb{R}^n$ is the input of the neural network. $\hat{y} \in \mathbb{R}^m$ is the output of the neural network. $\sigma : \mathbb{R} \mapsto \mathbb{R}$ is a non-linear activation function. $W_1 \in \mathbb{R}^{n_h \times n}$, $W_2 \in \mathbb{R}^{n_h \times n_h}$, $W_3 \in \mathbb{R}^{m \times n_h}$. $b_1 \in \mathbb{R}^{n_h}$, $b_2 \in \mathbb{R}^{n_h}$, $b_3 \in \mathbb{R}^m$. Sets of weights and biases are $W^3 = \{W_1, W_2, W_3\}$ and $b^3 = \{b_1, b_2, b_3\}$.

Similar to linear regression, we consider finding the parameters of neural networks as an optimization problem. This process is often called *training* in deep learning community. For example, for a two hidden layer networks $L = 3$, the corresponding minimization problem of finding the $f(x, W_*, b_*)$ can be formulated as,

$$W_*, b_* = \arg\min_{W^3, b^3} \frac{1}{M} \sum_i \| f(x^i; W^3, b^3) - y^i \|_2^2. \tag{B.4}$$

B.1.2 Training Neural Networks

Unfortunately, solving above eq. (B.4) becomes a formidable non-convex optimization problem. Currently, there is no established algorithm that guarantees global optimality. Therefore, local optimality is pursued with the help of stochastic gradient descent.

The simplest first-order gradient descent algorithm for minimizing a loss function $\text{Loss}(\theta; \mathcal{D})$ is,

$$\theta_{n+1} = \theta_n - \alpha \nabla_\theta \text{Loss}(\theta_n, \mathcal{D}), \tag{B.5}$$

where α is the learning rate and $\mathcal{D}$ is the training data.

For large datasets, directly performing the above algorithm can be extremely expensive as computing the loss function would require to go through every single sample at each step for updating the parameters. However, we only care about gradient, which is "averaged" over all samples. Naturally, one can subsample the whole data $\mathcal{D}$ with $\mathcal{D}_{\text{batch}}$ to have an unbiased estimator of the gradient. Then one can elongate the training iterations by going over each small patch of the whole dataset. This is called *stochastic gradient descent* algorithm, where $\nabla_\theta \text{Loss}(\theta_n, \mathcal{D})$ is replaced with $\nabla_\theta \text{Loss}(\theta_n, \mathcal{D}_{\text{batch}})$. It has been shown that the "randomness" introduced by subsampling the data helps the training of neural network to avoid the saddle points and converge to the local minimizer (*Jin et al.*, 2017).

Most of the popular algorithms in training deep neural nets are first-order opti-

mizers, primarily due to the efficiency of optimizing millions of parameters on huge datasets. Overfitting and poor generalization is often the major concern in deep learning rather than poor training errors. Therefore, higher order optimization methods become less popular. Empirically speaking, the best first order optimizer is Adam optimizer (*Kingma and Ba*, 2014), which can be thought as introducing a mechanism of adapting the learning rate in eq. (B.5) with the past information on trajectory of gradient dynamics.

B.1.3 Universal Approximator Property

Note that the key of deep learning is to learn high dimensional and complicated features, which is typically referred to as "representation learning". Naturally, it triggers a question for the neural network: *is the feedforward neural network structure capable of describing arbitrarily complex features*? Fortunately, universal approximator property of neural networks confirms the power of neural nets in describing arbitrarily complex high-dimensional functions.

One of the most prominent results came from *Hornik et al.* (1989). They start from a special class of FNN with *squashing function* defined as below:

Definition B.1 (Squashing function)**.** A function $G : \mathbb{R} \mapsto [0, 1]$ is a squashing function if it is non-decreasing and limited to 0 at $-\infty$ and 1 at ∞.

As stated in theorem B.2, they show that there is always a member in the family of a single hidden layer FNN with squashing function with *arbitrary hidden layer width* that can approximate any Borel measurable function arbitrarily well in any measure.

Theorem B.2. *For all squashing function G and all probability measure μ on $(\mathbb{R}^r, \mathcal{B}^r)$, $\Sigma^r G$ is $\rho_\mu-$dense in $\mathcal{M}^r$.*

Note that $\Sigma^r G$ is a notation for a single hidden layer FNN with G as non-linear activation that maps from $\mathbb{R}^r$ to $\mathbb{R}$. $\mathcal{B}^r$ is the Borel σ-field of $\mathbb{R}^r$. $\mathcal{M}^r$ is a set of all

Borel measurable function $\mathbb{R}^r \mapsto \mathbb{R}$. $\rho_\mu - dense$ implies two functions in each sets are likely to be close to each other[1]. For example, theorem B.2 indicates *sigmoid* is a universal approximator:

$$\sigma_{\text{sigmoid}}(x) = \frac{1}{1+e^{-x}}. \tag{B.6}$$

Another important work came from the famous work by *Chen and Chen* (1995) on learning non-linear *operators* with universal approximator. Surprisingly, they show that *any continuous activation function that is not a polynomial*, is qualified to be a non-linear activation that guarantees universal approximator property for FNN. For example, eLU activation function (*Clevert et al.*, 2015) falls in such category:

$$\sigma_{\text{eLU}}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{otherwise} \end{cases}, \tag{B.7}$$

where α is a hyperparameter that controls the amount of activation for negative input.

Recently, studies on leveraging finite-width but increasing depth of layers have been reported. Notably, *Perekrestenko et al.* (2018) proved finite-width deep ReLU networks are also universal approximators. However, there is a bound on the width imposed by the topological constraints (*Johnson*, 2018).

B.1.4 How to efficiently train a neural network?

As a highly non-linear non-convex optimization problem, deep neural nets are well-known for its difficulty to train, i.e., to obtain small training/validation error. Empirically, we found several useful tricks to obtain a good model:

- *"scale" of the input and output matters.* One should perform standard normalization on both inputs x and outputs y. The standard normalization is to

[1] $\forall g \in \mathcal{M}^r$ one can always find a sequence of function f_n s.t. $\lim_{n\to\infty} \rho_\mu(f_n, g) = 0$, where μ defines a metric ρ_μ from $\mathcal{M}^r \times \mathcal{M}^r$ to $\mathbb{R}^+$: for $f, g \in \mathcal{M}^r$, $\rho_\mu(f,g) = inf\{\epsilon > 0 : \mu\{x : |f(x) - g(x)| > \epsilon\} < \epsilon\}$, which says the area of the region where f and g differs is larger than a certain threshold.

subtract its sample mean and then divide by its sample standard deviation. This is because the parameters is often initialized between -1 and 1, which is data-independent.

- *initialization matters.* Usually, truncated normal distribution from `Tensorflow` as `tf.compat.v1.truncated_normal_initializer` performs very well.

- *size of batch matters.* Size of batch determines the number of subsampled data in a gradient update for the weights. The smaller the batch, the faster the training process but the higher the variance in the gradient so it might be fast in reducing the loss initially but at later stage, one can observe a substantial amount of fluctuations in the learning curve. As a rule of thumb, it is recommended to choose batch size between 16 and 512 depending on the size of the training data.

- *optimizer matters.* Usually Adam optimizer (*Kingma and Ba*, 2014) performs the best in training.

- *activation matters.* Swish activation (*Ramachandran et al.*, 2017) $x \cdot \sigma_{\text{sigmoid}}(x)$ usually performs well. Sometimes penalized tanh (*Xu et al.*, 2016),

$$\sigma_{\text{ptanh}}(x) = \begin{cases} \sigma_{\text{tanh}}(x), & \text{if } x > 0, \\ 0.25\sigma_{\text{tanh}}(x), & \text{otherwise,} \end{cases}, \tag{B.8}$$

  also performs good.

- *randomly shuffling data matters for mini-batch training.* Note that mini-batch training requires subsampling a subset from the whole training data. In order to obtain an unbiased estimator of the true gradient, randomly shuffling the data before training is necessary.

- *monitoring learning curve during training.* Although one doesn't have to per-

form cross validation in training deep learning models, it is suggested to randomly split out a small fraction of data (usually 5% to 10%) as validation data. Then, model performance on these data will be monitored to inform the selection of *hyperparameter*.

- *trial-and-error searching for best architectures.* Such architecture includes the number of units in each layer, type of activations, etc. They are called *hyperparameters* in machine learning community. The simplest way to find a good architecture is to start randomly choosing such hyperparameters then train repeatedly and only save the best model, e.g., *Lusch et al.* (2018b). For interested readers, there is a large amount of related literature with more sophisticated ways to find the optimal architecture under the topic of *Auto-ML* (*Hutter et al.*, 2019).

B.1.5 How to choose a deep learning platform?

There are two most popular platforms in `Python` developing deep learning models: `Tensorflow` and `PyTorch`. The former also has two versions `v1` and `v2`. `Tensorflow v1` is based on *static* computational graph (construct the graph then execute many times) while `Tensorflow v2` and `PyTorch` are based on *dynamic* computational graph. In terms of research and development, we recommend either using `Tensorflow v1` or `PyTorch`. The former is a bit more difficult to learn at the beginning. However, experienced user can easily construct any customized deep learning model easily. While the latter is easier for beginners and has better supports from deep learning research community (e.g., more off-the-shelf packages of state-of-the-art algorithms). However, it can be slow in execution due to the dynamic nature. Nevertheless, `PyTorch` has so far the best `C++` support because one can almost write the same amount of code with similar syntax in `C++` to build a standard deep learning model.

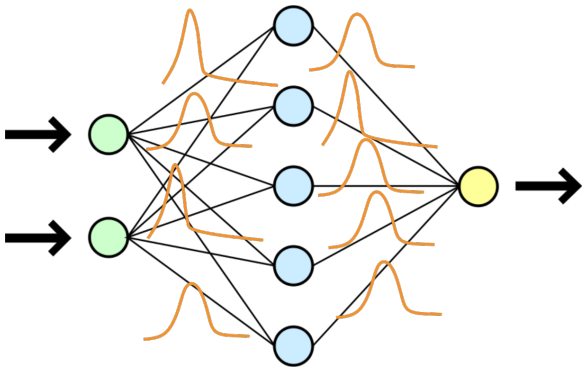


Figure B.1: Illustration of Bayesian neural network with two inputs and one output.

B.2 Bayesian Deep Learning

One of the most critical concerns from science and engineering community for deep learning is *safety*: for any deep learning model implemented in real life, it is natural to ask: how much confidence does the black-box model have for a certain input? One straightforward idea originated from scalable Bayesian inference community (*Neal*, 2012; *MacKay*, 1992) is to combine Bayesian inference with deep learning to construct Bayesian neural networks.

Recall that in appendix A.2, we have described ADVI, which is a powerful scalable Bayesian framework with automatic differentiation. Consequently, one can establish a Bayesian deep learning (BDL) model with ADVI. As illustrated in fig. B.1, instead of being *deterministic* parameters, the weights and biases in the model now become *probabilistic* distributions. Therefore, one can simply sample weights and biases from the distribution and perform Monte Carlo approximation to obtain the distribution of output given a certain input.

In this thesis, we establish Bayesian deep learning models with the ADVI. For readers that are reluctant to perform the aforementioned variational inference on the deep neural nets, there are several other easier options:

- *Monte Carlo Dropouts* (*Gal and Ghahramani*, 2016). One just need to train a

neural network with dropout regularization then the resulting model is simply an approximated variational inference for deep neural nets under mild assumption,

- *Laplace Approximation* (*Ritter et al.*, 2018),
- *Deep Ensembles* (*Lakshminarayanan et al.*, 2017).

B.2.1 How to efficiently train a Bayesian deep neural net with ADVI?

In practice, performing the ADVI above on neural networks is not as efficient as performing the deterministic counterpart. For a typical application in fluid dynamics, it would take five times the CPU time. Therefore, it is recommended to start with training with a small variance on the distribution of model parameters, i.e., let the mean converges first and then optimize for the variance. Alternatively, one can equivalently train a deterministic network then save the parameters as initial weights for distribution mean of the Bayesian neural nets.

APPENDIX C

Computational complexity of SPKDMD

Computational cost of the proposed framework can be divided in three parts:

1. Distributed SVD
2. KDMD/EDMD algorithms
3. Multi-task feature learning with the parameter sweep (solving eq. (4.9) with different values of α)

The computational complexity for each step in the algorithm is summarized in table C.1, where n is the dimension of the system state, M is the number of snapshots in the training [1], r is the rank of the reduced system after SVD, $\hat{L}$ is the user-defined cut off for "accurate" features, and N_{iter} is the maximal number of iterations user-defined to achieve a residual threshold, e.g. 10^{-12}.

As shown in the Table. C.1, the theoretical computational complexity for multi-task ElasticNet with an α sweep is $O(N_\alpha N_{iter} \hat{L}^2 r)$. Note that this is a worst case simply because - except for the first α - we *reuse the previous optimal solution* as the initial condition for the new objective function to start the iterative optimization

[1] for conciseness, we assume the number of training snapshots equals the number of validation snapshots.

Table C.1: Computational complexity of each step in the proposed sparsity-promoting framework.

Step	Computational complexity
SVD (QR iteration with vectors)	$O(Mn^2)$ (*Dongarra et al.*, 2018)
KDMD	$O(M^3)$
error evaluation & pruning	$O(M^2 n)$
multi-task ElasticNet with a fixed penalty coef. α	$O(N_{iter}\hat{L}^2 r)$
multi-task ElasticNet with N_α coefs. (worst case)	$O(N_\alpha N_{iter}\hat{L}^2 r)$

process. Also, thanks to SVD-based dimension reduction, the cost scales linearly with the reduced system rank r. Moreover, the user-defined linearly evolving error truncation $\hat{L}$ helps reduce that complexity as well instead of scaling with the number of snapshots M. Lastly, there is a cubic theoretical complexity associated with the number of snapshots when applying KDMD. The number of snapshots in a typical high fidelity simulation is $O(10^3)$. That is to say, $r < 10^3$ and $\hat{L} < 10^3$. We note that computational efficiency can be further improved, but this will be left for future work.

APPENDIX D

Hyperparameter selection for SPKDMD

D.1 2D fixed point attractor

We perform grid search in parallel for the selection of σ and the truncated rank r over the range: $\sigma \in [10^{-1}, 10]$ with 80 points uniformly distributed in the log sense and $r = 36, 50, 70$ to find a proper combination of r and σ. As shown in fig. D.1, the higher rank r leads to larger number of linearly evolving eigenfunctions. Thus, it is more crucial to choose a proper scale σ than r from fig. D.1. However, considering the simplicity of this problem, $\sigma = 2$ and $r = 36$ would suffice.

D.2 Cylinder flow case

As the flow is not turbulent, we choose hyperparameters for the $Re = 70$ case and fix them for all the other cases. We sweep σ from $[1, 10^5]$ with 30 points uniformly distributed in the log sense and $r = 120, 140, 160, 180, 200$ as shown in figure D.2. From the plot, we choose $r = 180$ and $\sigma = 3$ for an approximate choice of the hyperparameter. Again, we observe that the number of accurate eigenfunctions first increases then decreases with σ increasing and the saturation of rank truncation at

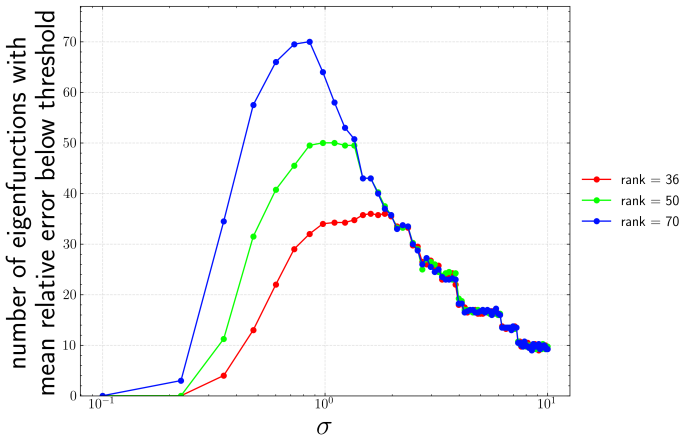


Figure D.1: Hyperparameter search for isotropic Gaussian KDMD on the 2D fixed point attractor.

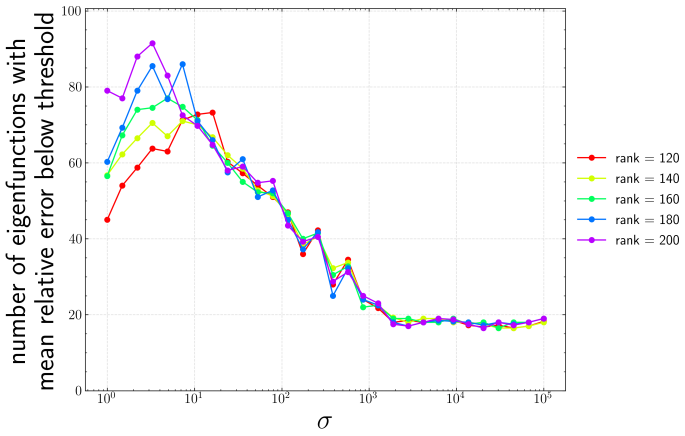


Figure D.2: Hyperparameter search for isotropic Gaussian KDMD on transient cylinder wake flows.

higher σ which is related to the variation in the characteristic scale of the features with respect to σ.

D.3 Turbulent Ship-airwake case

Grid search in parallel for the selection of σ and r is performed over the range $\sigma \in [1, 10^5]$ with 50 points uniformly distributed in the log sense, $r = 40, 80, 120, 130$. As shown in fig. D.3, a good choice of σ can be 200 for the case of $\alpha_\infty = 5°$. Note

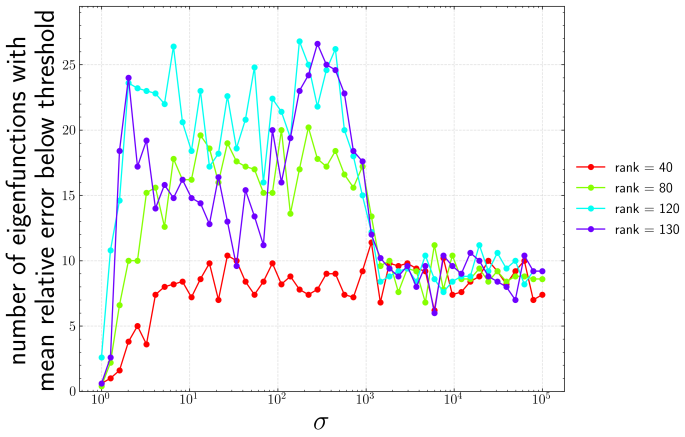


Figure D.3: Hyperparameter search for isotropic Gaussian KDMD on transient ship airwake.

that since the hyperparameter selection is performed with a 5-fold cross validation on the training data, we only have upto $166 * 0.8 \approx 132$ data points, i.e., maximal possible rank is 132. While in the actual training, we have maximal rank up to 166. Note that as long as the system is well-conditioned, the higher the rank, the richer the subspace. Here we take $\sigma = 200$ and $r = 135$.

APPENDIX E

Property of negative semi-definite matrix

Definition E.1. An $n \times n$ real matrix (and possibly non-symmetric) $\mathbf{A}$ is called negative semi-definite, if $\mathbf{x}^\top \mathbf{A}\mathbf{x} \leqslant 0$, for all non-zero vectors $\mathbf{x} \in \mathbb{R}^{n\times 1}$.

Lemma E.2. *For a negative semi-definite real square matrix, the real part of all of its eigenvalues is non-positive.*

Proof. Consider the general negative semi-definite matrix defined in definition E.1 as $\mathbf{A} \in \mathbb{R}^{n\times n}$, such that $\mathbf{x}^\top \mathbf{A}\mathbf{x} \leqslant 0$ for any non-zero vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$. Without loss of generality, denote its eigenvalues as $\lambda = \alpha + j\beta$, where $\alpha, \beta \in \mathbb{R}$ and corresponding eigenvectors as $\mathbf{v} = \mathbf{v}_r + j\mathbf{v}_i$ where $\mathbf{v}_r, \mathbf{v}_i \in \mathbb{R}^n$.

Then we have $0 = (\mathbf{A} - \lambda)\mathbf{v} = (\mathbf{A} - \alpha - j\beta)(\mathbf{v}_r + j\mathbf{v}_i)$, which further leads to $(\mathbf{A} - \alpha)\mathbf{v}_r = -\beta\mathbf{v}_i$ and $(\mathbf{A} - \alpha)\mathbf{v}_i = \beta\mathbf{v}_r$. Then we have $\mathbf{v}_r^\top(\mathbf{A} - \alpha)\mathbf{v}_r = -\beta\mathbf{v}_i^\top\mathbf{v}_r$, and $\mathbf{v}_i^\top(\mathbf{A} - \alpha)\mathbf{v}_i = \mathbf{v}_i^\top\beta\mathbf{v}_r$. Thus adding the two equations, we have $\mathbf{v}_r^\top(\mathbf{A} - \alpha)\mathbf{v}_r + \mathbf{v}_i^\top(\mathbf{A} - \alpha)\mathbf{v}_i = 0$, which implies $\alpha = (\mathbf{v}_r^\top\mathbf{A}\mathbf{v}_r + \mathbf{v}_i^\top\mathbf{A}\mathbf{v}_i)/(\mathbf{v}_r^\top\mathbf{v}_r + \mathbf{v}_i^\top\mathbf{v}_i)$.

Using the definition of negative semi-definite matrices, we have $\mathbf{v}_r^\top\mathbf{A}\mathbf{v}_r \leqslant 0$ and $\mathbf{v}_i^\top\mathbf{A}\mathbf{v}_i \leqslant 0$, even if $\mathbf{v}_r$ or $\mathbf{v}_i = \mathbf{0}$. Since there is at least one non-zero vector between $\mathbf{v}_r$ and $\mathbf{v}_i$, one can safely arrive at $\alpha \leqslant 0$ that the real part of any eigenvalue of $\mathbf{A}$ is non-positive. □

BIBLIOGRAPHY

BIBLIOGRAPHY

Abadi, M., et al. (2016), Tensorflow: A system for large-scale machine learning, in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283.

Abarbanel, H. D., R. Brown, J. J. Sidorowich, and L. S. Tsimring (1993), The analysis of observed chaotic data in physical systems, *Reviews of modern physics*, *65*(4), 1331.

Abraham, I., G. De La Torre, and T. D. Murphey (2017), Model-based control using koopman operators, *arXiv preprint arXiv:1709.01568*.

Alessandri, A., P. Bagnerini, M. Gaggero, D. Lengani, and D. Simoni (2019), Dynamic mode decomposition for the inspection of three-regime separated transitional boundary layers using a least squares method, *Physics of Fluids*, *31*(4), 044,103.

Anderson, E., et al. (1999), *LAPACK Users' Guide*, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA.

Anderson Jr, J. D. (2010), *Fundamentals of aerodynamics*, Tata McGraw-Hill Education.

Ansys (2016), Ansys fluent user's guide, release 17.0, *Ansys*.

Arbabi, H., and I. Mezic (2017), Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator, *SIAM Journal on Applied Dynamical Systems*, *16*(4), 2096–2126.

Arbabi, H., and I. Mezić (2017), Study of dynamics in post-transient flows using koopman mode decomposition, *Physical Review Fluids*, *2*(12), 124,402.

Arbabi, H., M. Korda, and I. Mezic (2018a), A data-driven koopman model predictive control framework for nonlinear flows (2018), *arXiv preprint arXiv:1804.05291*.

Arbabi, H., M. Korda, and I. Mezic (2018b), A data-driven koopman model predictive control framework for nonlinear flows, *arXiv preprint arXiv:1804.05291*.

Argyriou, A., T. Evgeniou, and M. Pontil (2008a), Convex multi-task feature learning, *Machine Learning*, *73*(3), 243–272.

Argyriou, A., M. Pontil, Y. Ying, and C. A. Micchelli (2008b), A spectral regularization framework for multi-task structure learning, in *Advances in neural information processing systems*, pp. 25–32.

Arnold-Medabalimi, N., C. Huang, and K. Duraisamy (2020), Data-driven modal decomposition techniques for high-dimensional flow fields, in *Data Analysis for Direct Numerical Simulations of Turbulent Combustion*, pp. 135–155, Springer.

Aronszajn, N. (1950), Theory of reproducing kernels, *Transactions of the American mathematical society*, *68*(3), 337–404.

Arrowsmith, D., and C. M. Place (1992), *Dynamical systems: differential equations, maps, and chaotic behaviour*, vol. 5, CRC Press.

Askham, T., and J. N. Kutz (2018), Variable projection methods for an optimized dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, *17*(1), 380–416.

Attinger, E., A. Anne, and D. McDonald (1966), Use of Fourier series for the analysis of biological systems, *Biophysical Journal*, *6*(3), 291.

Azencot, O., W. Yin, and A. Bertozzi (2019), Consistent dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, *18*(3), 1565–1585.

Bach, F., R. Jenatton, J. Mairal, G. Obozinski, et al. (2012), Optimization with sparsity-inducing penalties, *Foundations and Trends® in Machine Learning*, *4*(1), 1–106.

Bagheri, S. (2013), Koopman-mode decomposition of the cylinder wake, *Journal of Fluid Mechanics*, *726*, 596–623.

Baudin, M., et al. (2015), pydoe.

Bazán, F. S. (2000), Conditioning of rectangular Vandermonde matrices with nodes in the unit disk, *SIAM Journal on Matrix Analysis and Applications*, *21*(2), 679–693.

Berkooz, G., P. Holmes, and J. L. Lumley (1993), The proper orthogonal decomposition in the analysis of turbulent flows, *Annual review of fluid mechanics*, *25*(1), 539–575.

Berman, L., and A. Feuer (2007), On perfect conditioning of Vandermonde matrices on the unit circle, *Electronic Journal of Linear Algebra*, *16*(1), 13.

Billings, S. A. (2013), *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*, John Wiley & Sons.

Björck, A., and V. Pereyra (1970), Solution of Vandermonde systems of equations, *Mathematics of Computation*, *24*(112), 893–903.

Blei, D. M., M. I. Jordan, et al. (2006), Variational inference for dirichlet process mixtures, *Bayesian analysis*, *1*(1), 121–143.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017), Variational inference: A review for statisticians, *Journal of the American Statistical Association*, *112*(518), 859–877.

Blundell, C., J. Cornebise, K. Kavukcuoglu, and D. Wierstra (2015), Weight uncertainty in neural networks, *arXiv preprint arXiv:1505.05424*.

Box, G. E., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung (2015), *Time series analysis: forecasting and control*, John Wiley & Sons.

Boyd, J. P. (2001), *Chebyshev and Fourier spectral methods*, Courier Corporation.

Broomhead, D. S., and R. Jones (1989), Time-series analysis, *Proc. R. Soc. Lond. A*, *423*(1864), 103–121.

Broomhead, D. S., and G. P. King (1986), Extracting qualitative dynamics from experimental data, *Physica D: Nonlinear Phenomena*, *20*(2-3), 217–236.

Bruder, D., B. Gillespie, C. D. Remy, and R. Vasudevan (2019), Modeling and control of soft robots using the koopman operator and model predictive control, *arXiv preprint arXiv:1902.02827*.

Brunton, S. L., and J. N. Kutz (2019), *Data-driven science and engineering: Machine learning, dynamical systems, and control*, Cambridge University Press.

Brunton, S. L., J. L. Proctor, and J. N. Kutz (2013), Compressive sampling and dynamic mode decomposition, *arXiv preprint arXiv:1312.5186*.

Brunton, S. L., B. W. Brunton, J. L. Proctor, and J. N. Kutz (2016a), Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control, *PloS one*, *11*(2), e0150,171.

Brunton, S. L., J. L. Proctor, and J. N. Kutz (2016b), Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences*, p. 201517384.

Brunton, S. L., B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz (2017), Chaos as an intermittently forced linear system, *Nature communications*, *8*(1), 19.

Budišić, M., and I. Mezić (2012), Geometry of the ergodic quotient reveals coherent structures in flows, *Physica D: Nonlinear Phenomena*, *241*(15), 1255–1269.

Budišić, M., R. Mohr, and I. Mezić (2012), Applied koopmanism, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *22*(4), 047,510.

Bulat, G., W. Jones, and A. Marquis (2013), Large eddy simulation of an industrial gas-turbine combustion chamber using the sub-grid pdf method, *Proceedings of the combustion institute*, *34*(2), 3155–3164.

Candes, E. J., and T. Tao (2006), Near-optimal signal recovery from random projections: Universal encoding strategies?, *IEEE transactions on information theory*, *52*(12), 5406–5425.

Cao, L. (1997), Practical method for determining the minimum embedding dimension of a scalar time series, *Physica D: Nonlinear Phenomena*, *110*(1-2), 43–50.

Carlberg, K., C. Farhat, J. Cortial, and D. Amsallem (2013), The gnat method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, *Journal of Computational Physics*, *242*, 623–647.

Champion, K. P., S. L. Brunton, and J. N. Kutz (2019), Discovery of nonlinear multiscale systems: Sampling strategies and embeddings, *SIAM Journal on Applied Dynamical Systems*, *18*(1), 312–333.

Chang, B., L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham (2018), Reversible architectures for arbitrarily deep residual neural networks, in *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chang, B., M. Chen, E. Haber, and E. H. Chi (2019), Antisymmetricrnn: A dynamical system view on recurrent neural networks, *arXiv preprint arXiv:1902.09689*.

Chapman, D. R. (1979), Computational aerodynamics development and outlook, *AIAA journal*, *17*(12), 1293–1313.

Chen, K. K., J. H. Tu, and C. W. Rowley (2012), Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses, *Journal of nonlinear science*, *22*(6), 887–915.

Chen, T., and H. Chen (1995), Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Transactions on Neural Networks*, *6*(4), 911–917.

Cho, Y., and L. K. Saul (2009), Kernel methods for deep learning, in *Advances in neural information processing systems*, pp. 342–350.

Chollet, F., et al. (2015), Keras, `https://keras.io`.

Chorin, A. J., and O. H. Hald (2014), Estimating the uncertainty in underresolved nonlinear dynamics, *Mathematics and Mechanics of Solids*, *19*(1), 28–38.

Clevert, D.-A., T. Unterthiner, and S. Hochreiter (2015), Fast and accurate deep network learning by exponential linear units (elus), *arXiv preprint arXiv:1511.07289*.

Commons, W. (2020a), File:aerion sbj model.jpg — wikimedia commons, the free media repository, [Online; accessed 23-August-2020].

Commons, W. (2020b), File:launch of falcon 9 carrying crs-6 dragon (17170624642).jpg — wikimedia commons, the free media repository, [Online; accessed 23-August-2020].

Córdova, A., W. Gautschi, and S. Ruscheweyh (1990), Vandermonde matrices on the circle: spectral properties and conditioning, *Numerische Mathematik*, *57*(1), 577–591.

Dalcin, L. D., R. R. Paz, P. A. Kler, and A. Cosimo (2011), Parallel distributed computing using python, *Advances in Water Resources*, *34*(9), 1124–1139.

Dawson, S. T., M. S. Hemati, M. O. Williams, and C. W. Rowley (2016), Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition, *Experiments in Fluids*, *57*(3), 42.

Deem, E. A., L. N. Cattafesta III, M. S. Hemati, H. Zhang, C. Rowley, and R. Mittal (2020), Adaptive separation control of a laminar boundary layer using online dynamic mode decomposition, *Journal of Fluid Mechanics*, *903*, A21.

DeGennaro, A. M., and N. M. Urban (2019), Scalable extended dynamic mode decomposition using random kernel approximation, *SIAM Journal on Scientific Computing*, *41*(3), A1482–A1499.

Denker, J. S., and Y. Lecun (1991), Transforming neural-net output levels to probability distributions, in *Advances in neural information processing systems*, pp. 853–859.

Deyle, E. R., and G. Sugihara (2011), Generalized theorems for nonlinear state space reconstruction, *PLoS One*, *6*(3).

Dongarra, J., M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Tomov, and I. Yamazaki (2018), The singular value decomposition: Anatomy of optimizing an algorithm for extreme scale, *SIAM review*, *60*(4), 808–865.

Donoho, D. L. (2006), Compressed sensing, *IEEE Transactions on information theory*, *52*(4), 1289–1306.

Dowell, E. H., and K. C. Hall (2001), Modeling of fluid-structure interaction, *Annual review of fluid mechanics*, *33*(1), 445–490.

Drazin, P. G., and W. H. Reid (2004), *Hydrodynamic stability*, Cambridge university press.

Drmac, Z., I. Mezic, and R. Mohr (2019), Data driven Koopman spectral analysis in Vandermonde–Cauchy form via the DFT: Numerical method and theoretical insights, *SIAM Journal on Scientific Computing*, *41*(5), A3118–A3151.

Eivazi, H., L. Guastoni, P. Schlatter, H. Azizpour, and R. Vinuesa (2020), Recurrent neural networks and koopman-based frameworks for temporal predictions in turbulence, *arXiv preprint arXiv:2005.02762*.

Erichson, N. B., M. Muehlebach, and M. W. Mahoney (2019), Physics-informed autoencoders for lyapunov-stable fluid flow prediction, *arXiv preprint arXiv:1905.10866*.

Fey, U., M. König, and H. Eckelmann (1998), A new strouhal–reynolds-number relationship for the circular cylinder in the range 47¡ re¡ 2× 10 5, *Physics of Fluids*, *10*(7), 1547–1549.

Forrest, J. S., and I. Owen (2010), An investigation of ship airwakes using detached-eddy simulation, *Computers & Fluids*, *39*(4), 656–673.

Frank, R. J., N. Davey, and S. P. Hunt (2001), Time series prediction and neural networks, *Journal of intelligent and robotic systems*, *31*(1-3), 91–103.

Friedman, J., T. Hastie, and R. Tibshirani (2001), *The elements of statistical learning*, vol. 1, Springer series in statistics New York.

Froyland, G., G. A. Gottwald, and A. Hammerlindl (2014), A computational method to extract macroscopic variables and their dynamics in multiscale systems, *SIAM Journal on Applied Dynamical Systems*, *13*(4), 1816–1846.

Gal, Y., and Z. Ghahramani (2016), Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in *international conference on machine learning*, pp. 1050–1059.

Galioto, N., and A. Gorodetsky (2020), Bayesian system id: Optimal management of parameter, model, and measurement uncertainty, *arXiv preprint arXiv:2003.02359*.

Gautschi, W. (1990), How (un) stable are Vandermonde systems, *Asymptotic and computational analysis*, *124*, 193–210.

Gavish, M., and D. L. Donoho (2014), The optimal hard threshold for singular values is $4/\sqrt{3}$, *IEEE Transactions on Information Theory*, *60*(8), 5040–5053.

Gelman, A. (2004), Prior distributions for variance parameters in hierarchical models, *Econometrics*.

Gibson, J. F., J. Doyne Farmer, M. Casdagli, and S. Eubank (1992), An analytic approach to practical state space reconstruction, *Physica. D, Nonlinear phenomena*, *57*(1-2), 1–30.

Gilmore, R., and M. Lefranc (2011), *The Topology of Chaos Alice in Stretch and Squeezeland*, 2nd ed. ed., John Wiley & Sons, Ltd.

Gin, C., B. Lusch, S. L. Brunton, and J. N. Kutz (2019), Deep learning models for global coordinate transformations that linearize pdes, *arXiv preprint arXiv:1911.02710*.

Goodfellow, I., Y. Bengio, and A. Courville (2016), *Deep learning*, MIT press.

Gouasmi, A., E. J. Parish, and K. Duraisamy (2017), A priori estimation of memory effects in reduced-order models of nonlinear systems using the mori–zwanzig formalism, *Proc. R. Soc. A*, *473*(2205), 20170,385.

Govindarajan, N., H. Arbabi, L. van Blargian, T. Matchen, E. Tegling, et al. (2016), An operator-theoretic viewpoint to non-smooth dynamical systems: Koopman analysis of a hybrid pendulum, in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 6477–6484, IEEE.

Graves, A. (2011), Practical variational inference for neural networks, in *Advances in neural information processing systems*, pp. 2348–2356.

Gruyitch, L. T. (2018), *Observability and Controllability of General Linear Systems*, CRC Press.

Guckenheimer, J., and P. Holmes (2013), *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, vol. 42, Springer Science & Business Media.

Haber, E., and L. Ruthotto (2017), Stable architectures for deep neural networks, *Inverse Problems*, *34*(1), 014,004.

Hao, W., and Y. Han (2020), Data driven control with learned dynamics: Model-based versus model-free approach, *arXiv preprint arXiv:2006.09543*.

Harvazinski, M. E., C. Huang, V. Sankaran, T. W. Feldman, W. E. Anderson, C. L. Merkle, and D. G. Talley (2015), Coupling between hydrodynamics, acoustics, and heat release in a self-excited unstable combustor, *Physics of Fluids*, *27*(4), 045,102.

Haseli, M., and J. Cortés (2019), Efficient identification of linear evolutions in nonlinear vector fields: Koopman invariant subspaces, *arXiv preprint arXiv:1909.01419*.

He, K., X. Zhang, S. Ren, and J. Sun (2016), Deep residual learning for image recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, S., E. Jonsson, C. A. Mader, and J. Martins (2019), A coupled newton-krylov time spectral solver for wing flutter and lco prediction, in *AIAA Aviation 2019 Forum*, p. 3549.

Herstein, I. N. (1975), *Topics in algebra*, Xerox College Pub.

Hinton, G., and D. Van Camp (1993), Keeping neural networks simple by minimizing the description length of the weights, in *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*, Citeseer.

Hirsch, M. W., S. Smale, and R. L. Devaney (2012), *Differential equations, dynamical systems, and an introduction to chaos*, Academic press.

Hochreiter, S., and J. Schmidhuber (1997), Long short-term memory, *Neural computation*, *9*(8), 1735–1780.

Holmes, P., J. L. Lumley, G. Berkooz, and C. W. Rowley (2012), *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge university press.

Hornik, K., M. Stinchcombe, H. White, et al. (1989), Multilayer feedforward networks are universal approximators., *Neural networks*, *2*(5), 359–366.

Hua, J.-C., F. Noorian, D. Moss, P. H. Leong, and G. H. Gunaratne (2017), High-dimensional time series prediction using kernel-based koopman mode regression, *Nonlinear Dynamics*, *90*(3), 1785–1806.

Huang, C., W. E. Anderson, M. E. Harvazinski, and V. Sankaran (2016), Analysis of self-excited combustion instabilities using decomposition techniques, *AIAA Journal*, *54*(9), 2791–2807.

Huang, C., K. Duraisamy, and C. Merkle (2018), Challenges in reduced order modeling of reacting flows, in *2018 Joint Propulsion Conference*, p. 4675.

Huang, C., C. R. Wentland, K. Duraisamy, and C. Merkle (2020a), Model reduction for multi-scale transport problems using structure-preserving least-squares projections with variable transformation, *arXiv preprint arXiv:2011.02072*.

Huang, C., C. R. Wentland, K. Duraisamy, and C. Merkle (2020b), Model reduction for multi-scale transport problems using structure-preserving least-squares projections with variable transformation, *arXiv preprint arXiv:2011.02072*.

Hussain, A. K. M. F., and W. C. Reynolds (1970), The mechanics of an organized wave in turbulent shear flow, *Journal of Fluid Mechanics*, *41*(2), 241–258.

Hutter, F., L. Kotthoff, and J. Vanschoren (2019), *Automated machine learning: methods, systems, challenges*, Springer Nature.

Ishihara, T., K. Morishita, M. Yokokawa, A. Uno, and Y. Kaneda (2016), Energy spectrum in high-resolution direct numerical simulations of turbulence, *Physical Review Fluids*, *1*(8), 082,403.

Jasak, H., A. Jemcov, Z. Tukovic, et al. (2007), Openfoam: A c++ library for complex physics simulations, in *International workshop on coupled methods in numerical dynamics*, vol. 1000, pp. 1–20, IUC Dubrovnik Croatia.

Jin, C., R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan (2017), How to escape saddle points efficiently, *arXiv preprint arXiv:1703.00887*.

Johnson, J. (2018), Deep, skinny neural networks are not universal approximators, *arXiv preprint arXiv:1810.00393*.

Jones, E., T. Oliphant, P. Peterson, et al. (2001–), SciPy: Open source scientific tools for Python.

Jones, E., T. Oliphant, and P. Peterson (2014), {SciPy}: Open source scientific tools for {Python}.

Jovanović, M. R., P. J. Schmid, and J. W. Nichols (2014), Sparsity-promoting dynamic mode decomposition, *Physics of Fluids*, *26*(2), 024,103.

Juang, J.-N., and R. S. Pappa (1985), An eigensystem realization algorithm for modal parameter identification and model reduction, *Journal of guidance, control, and dynamics*, *8*(5), 620–627.

Jungers, R. M., and P. Tabuada (2019), Non-local linearization of nonlinear differential equations via polyflows, in *2019 American Control Conference (ACC)*, pp. 1–6, IEEE.

Kaiser, E., J. N. Kutz, and S. L. Brunton (2017), Data-driven discovery of koopman eigenfunctions for control, *arXiv preprint arXiv:1707.01146*.

Kamb, M., E. Kaiser, S. L. Brunton, and J. N. Kutz (2018), Time-delay observables for Koopman: Theory and applications, *arXiv preprint arXiv:1810.01479*.

Kennel, M. B., R. Brown, and H. D. Abarbanel (1992), Determining embedding dimension for phase-space reconstruction using a geometrical construction, *Physical review A*, *45*(6), 3403.

Khalil, H. K., and J. W. Grizzle (2002), *Nonlinear systems*, vol. 3, Prentice hall Upper Saddle River, NJ.

Kim, H., R. Eykholt, and J. Salas (1999), Nonlinear dynamics, delay times, and embedding windows, *Physica D: Nonlinear Phenomena*, *127*(1-2), 48–60.

Kingma, D. P., and J. Ba (2014), Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

Klenke, A. (2013), *Probability theory: a comprehensive course*, Springer Science & Business Media.

Klus, S., P. Koltai, and C. Schütte (2015), On the numerical approximation of the perron-frobenius and koopman operator, *arXiv preprint arXiv:1512.05997*.

Koopman, B., and J. v. Neumann (1932), Dynamical systems of continuous spectra, *Proceedings of the National Academy of Sciences*, *18*(3), 255–263.

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

Korda, M., and I. Mezić (2016), Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, *arXiv preprint arXiv:1611.03537*.

Korda, M., and I. Mezić (2018a), On convergence of extended dynamic mode decomposition to the koopman operator, *Journal of Nonlinear Science*, *28*(2), 687–710.

Korda, M., and I. Mezić (2018b), Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, *Automatica*, *93*, 149–160.

Kou, J., and W. Zhang (2017), An improved criterion to select dominant modes from dynamic mode decomposition, *European Journal of Mechanics-B/Fluids*, *62*, 109–129.

Kramer, B., P. Grover, P. Boufounos, S. Nabi, and M. Benosman (2017), Sparse sensing and dmd-based identification of flow regimes and bifurcations in complex flows, *SIAM Journal on Applied Dynamical Systems*, *16*(2), 1164–1196.

Kreindler, E., and P. Sarachik (1964), On the concepts of controllability and observability of linear systems, *IEEE Transactions on Automatic Control*, *9*(2), 129–136.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2017), Imagenet classification with deep convolutional neural networks, *Communications of the ACM*, *60*(6), 84–90.

Kucukelbir, A., D. Tran, R. Ranganath, A. Gelman, and D. M. Blei (2017), Automatic differentiation variational inference, *The Journal of Machine Learning Research*, *18*(1), 430–474.

Kundu, P. K., and I. M. Cohen (2001), *Fluid mechanics*, Elsevier.

Kunis, S., and D. Nagel (2018), On the condition number of Vandermonde matrices with pairs of nearly–colliding nodes, *arXiv preprint arXiv:1812.08645*.

Kutz, J. N., S. L. Brunton, B. W. Brunton, and J. L. Proctor (2016), *Dynamic mode decomposition: data-driven modeling of complex systems*, SIAM.

Lakshminarayanan, B., A. Pritzel, and C. Blundell (2017), Simple and scalable predictive uncertainty estimation using deep ensembles, in *Advances in neural information processing systems*, pp. 6402–6413.

Lan, Y., and I. Mezić (2013), Linearization in the large of nonlinear systems and koopman operator spectrum, *Physica D: Nonlinear Phenomena*, *242*(1), 42–53.

Landau, H. (1967), Sampling, data transmission, and the Nyquist rate, *Proceedings of the IEEE*, *55*(10), 1701–1706.

Landau, L. D. (1944), On the problem of turbulence, in *Dokl. Akad. Nauk USSR*, vol. 44, p. 311.

Le Clainche, S., and J. M. Vega (2017), Higher order dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, *16*(2), 882–925.

Leask, S., and V. McDonell (2020), Extraction of discrete spectra modes from video data using a deep convolutional koopman network, *arXiv preprint arXiv:2010.09245*.

Lee, K., and K. T. Carlberg (2020), Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, *Journal of Computational Physics*, *404*, 108,973.

Lee, S., and D. You (2019), Data-driven prediction of unsteady flow over a circular cylinder using deep learning, *Journal of Fluid Mechanics*, *879*, 217–254.

Leshno, M., V. Y. Lin, A. Pinkus, and S. Schocken (1993), Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, *Neural networks*, *6*(6), 861–867.

Li, Q., F. Dietrich, E. M. Bollt, and I. G. Kevrekidis (2017), Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *27*(10), 103,111.

Lietz, A. M., E. Johnsen, and M. J. Kushner (2017), Plasma-induced flow instabilities in atmospheric pressure plasma jets, *Applied Physics Letters*, *111*(11), 114,101.

Liu, F., G. S. Ng, and C. Quek (2007), RLDDE: A novel reinforcement learning-based dimension and delay estimator for neural networks in time series prediction, *Neurocomputing*, *70*(7-9), 1331–1341.

Liu, Q., and D. Wang (2016), Stein variational gradient descent: A general purpose bayesian inference algorithm, in *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, pp. 2378–2386, Curran Associates, Inc.

Liu, Q., and D. Wang (2018), Stein variational gradient descent as moment matching, *neural information processing systems*, pp. 8854–8863.

Lomax, R. G., and D. L. Hahs-Vaughn (2013), *Statistical concepts: A second course*, Routledge.

Lu, J., Y. Lu, and J. Nolen (2019), Scaling limit of the stein variational gradient descent: The mean field regime, *Siam Journal on Mathematical Analysis*, *51*(2), 648–671.

Luenberger, D. G. (1973), *Introduction to linear and nonlinear programming*, Addison-Wesley publishing company.

Lusch, B., J. N. Kutz, and S. L. Brunton (2018a), Deep learning for universal linear embeddings of nonlinear dynamics, *Nature communications*, *9*(1), 4950.

Lusch, B., J. N. Kutz, and S. L. Brunton (2018b), Deep learning for universal linear embeddings of nonlinear dynamics, *Nature Communications*, *9*(1), 4950, doi:10.1038/s41467-018-07210-0.

Lyu, Z., G. K. Kenway, C. Paige, and J. Martins (2013), Automatic differentiation adjoint of the reynolds-averaged navier-stokes equations with a turbulence model, in *21st AIAA Computational Fluid Dynamics Conference*, p. 2581.

MacKay, D. J. (1992), A practical bayesian framework for backpropagation networks, *Neural computation*, *4*(3), 448–472.

Mader, C. A., J. R. Martins, J. J. Alonso, and E. Van Der Weide (2008), Adjoint: An approach for the rapid development of discrete adjoint solvers, *AIAA journal*, *46*(4), 863–873.

Mamakoukas, G., M. Castano, X. Tan, and T. Murphey (2019), Local koopman operators for data-driven control of robotic systems, in *Robotics: science and systems*.

MATLAB (2010), *version 7.10.0 (R2010a)*, The MathWorks Inc., Natick, Massachusetts.

Maulik, R., A. Mohan, B. Lusch, S. Madireddy, P. Balaprakash, and D. Livescu (2020), Time-series learning of latent-space dynamics for reduced-order model closure, *Physica D: Nonlinear Phenomena*, *405*, 132,368.

Mauroy, A., and I. Mezić (2012), On the use of fourier averages to compute the global isochrons of (quasi) periodic dynamics, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *22*(3), 033,112.

Mauroy, A., and I. Mezić (2016), Global stability analysis using the eigenfunctions of the koopman operator, *IEEE Transactions on Automatic Control*, *61*(11), 3356–3369.

McGuinness, M. J. (1983), The fractal dimension of the lorenz attractor, *Physics Letters A*, *99*(1), 5–9.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979), Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, *21*(2), 239–245.

McKeon, B. J., and A. S. Sharma (2010), A critical-layer framework for turbulent pipe flow, *Journal of Fluid Mechanics*, *658*, 336–382.

McQuarrie, S. A., C. Huang, and K. Willcox (2020), Data-driven reduced-order models via regularized operator inference for a single-injector combustion process, *arXiv preprint arXiv:2008.02862*.

Mercer, B. (1909a), Xvi. functions of positive and negative type, and their connection the theory of integral equations, *Phil. Trans. R. Soc. Lond. A*, *209*(441-458), 415–446.

Mercer, J. (1909b), Functions of positive and negative type and their connection with the theory of integral equations, philosophical transsaction of the royal society of london, ser.

Meucci, A. (2009), Review of statistical arbitrage, cointegration, and multivariate ornstein-uhlenbeck, *Cointegration, and Multivariate Ornstein-Uhlenbeck (May 14, 2009)*.

Meyer, C. D. (2000), *Matrix analysis and applied linear algebra*, vol. 71, Siam.

Mezić, I. (2005), Spectral properties of dynamical systems, model reduction and decompositions, *Nonlinear Dynamics*, *41*(1-3), 309–325.

Mezić, I. (2013), Analysis of fluid flows via spectral properties of the koopman operator, *Annual Review of Fluid Mechanics*, *45*, 357–378.

Mezić, I. (2015), On applications of the spectral theory of the koopman operator in dynamical systems and control theory, in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 7034–7041, IEEE.

Mezić, I., and S. Wiggins (1999), A method for visualization of invariant sets of dynamical systems based on the ergodic partition, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *9*(1), 213–218.

Minh, H. Q. (2010), Some properties of gaussian reproducing kernel hilbert spaces and their implications for function approximation and learning theory, *Constructive Approximation*, *32*(2), 307–338.

Mohan, A., D. Daniel, M. Chertkov, and D. Livescu (2019), Compressed convolutional lstm: An efficient deep learning framework to model high fidelity 3d turbulence, *arXiv preprint arXiv:1903.00033*.

Moin, P., and K. Mahesh (1998), Direct numerical simulation: a tool in turbulence research, *Annual review of fluid mechanics*, *30*(1), 539–578.

Monkewitz, P. A., P. Huerre, and J.-M. Chomaz (1993), Global linear stability analysis of weakly non-parallel shear flows, *Journal of Fluid Mechanics*, *251*, 1–20.

Morton, J., F. D. Witherden, and M. J. Kochenderfer (2019), Deep variational koopman models: Inferring koopman observations for uncertainty-aware dynamics modeling and control, *arXiv preprint arXiv:1902.09742*.

Nathan Kutz, J., J. L. Proctor, and S. L. Brunton (2018), Applied koopman theory for partial differential equations and data-driven modeling of spatio-temporal systems, *Complexity*, *2018*.

Neal, R. M. (2012), *Bayesian learning for neural networks*, vol. 118, Springer Science & Business Media.

Nijmeijer, H., and A. Van der Schaft (1990), *Nonlinear dynamical control systems*, vol. 175, Springer.

Noack, B. R., K. Afanasiev, M. MORZYŃSKI, G. Tadmor, and F. Thiele (2003), A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *Journal of Fluid Mechanics*, *497*, 335–363.

Otto, S. E., and C. W. Rowley (2019a), Linearly recurrent autoencoder networks for learning dynamics, *SIAM Journal on Applied Dynamical Systems*, *18*(1), 558–593.

Otto, S. E., and C. W. Rowley (2019b), Linearly recurrent autoencoder networks for learning dynamics, *SIAM Journal on Applied Dynamical Systems*, *18*(1), 558–593.

Page, J., and R. R. Kerswell (2018), Koopman analysis of burgers equation, *Physical Review Fluids*, *3*(7), 071,901.

Pan, S., and N. Arnold-Medabalimi (2020), POD coefficients of 3D turbulent Rayleigh-Bénard convection at $Ra = 10^7$., `https://github.com/pswpswpsw/2020_Time_Delay_Paper_Rayleigh-Benard`.

Pan, S., and K. Duraisamy (2018a), Data-Driven Discovery of Closure Models, *SIAM Journal on Applied Dynamical Systems*, *17*(4), 2381–2413.

Pan, S., and K. Duraisamy (2018b), Data-driven discovery of closure models, *SIAM Journal on Applied Dynamical Systems*, *17*(4), 2381–2413.

Pan, S., and K. Duraisamy (2018c), Long-time predictive modeling of nonlinear dynamical systems using neural networks, *arXiv preprint arXiv:1805.12547*.

Pan, S., and K. Duraisamy (2020a), Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability, *SIAM Journal on Applied Dynamical Systems*, *19*(1), 480–509.

Pan, S., and K. Duraisamy (2020b), On the structure of time-delay embedding in linear models of non-linear dynamical systems, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *30*(7), 073,135.

Pan, S., and K. Duraisamy (2020c), Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability, *SIAM Journal on Applied Dynamical Systems*, *19*(1), 480–509.

Pan, S., and E. Johnsen (2017), The role of bulk viscosity on the decay of compressible, homogeneous, isotropic turbulence, *Journal of Fluid Mechanics*, *833*, 717–744.

Pan, S., N. Arnold-Medabalimi, and K. Duraisamy (2020), Sparsity-promoting algorithms for the discovery of informative Koopman invariant subspaces, *arXiv preprint arXiv:2002.10637*.

Pan, V. Y. (2016), How bad are Vandermonde matrices?, *SIAM Journal on Matrix Analysis and Applications*, *37*(2), 676–694.

Parikh, N., S. Boyd, et al. (2014), Proximal algorithms, *Foundations and Trends® in Optimization*, *1*(3), 127–239.

Parish, E. J., C. Wentland, and K. Duraisamy (2018), The Adjoint Petrov-Galerkin Method for Non-Linear Model Reduction, *arXiv e-prints*.

Parish, E. J., C. R. Wentland, and K. Duraisamy (2020), The adjoint petrov–galerkin method for non-linear model reduction, *Computer Methods in Applied Mechanics and Engineering*, *365*, 112,991.

Parry, W. (2004), *Topics in ergodic theory*, vol. 75, Cambridge university press.

Pedregosa, F., et al. (2011), Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, *12*, 2825–2830.

Peherstorfer, B., and K. Willcox (2016), Data-driven operator inference for nonintrusive projection-based model reduction, *Computer Methods in Applied Mechanics and Engineering*, *306*, 196–215.

Perekrestenko, D., P. Grohs, D. Elbrächter, and H. Bölcskei (2018), The universal approximation power of finite-width deep relu networks, *arXiv preprint arXiv:1806.01528*.

Petersen, K. B., M. S. Pedersen, et al. (2008), The matrix cookbook, *Technical University of Denmark*, *7*(15), 510.

Phan, M., L. G. Horta, J.-N. Juang, and R. W. Longman (1993), Linear system identification via an asymptotically stable observer, *Journal of Optimization Theory and Applications*, *79*(1), 59–86.

Pitsch, H. (2006), Large-eddy simulation of turbulent combustion, *Annu. Rev. Fluid Mech.*, *38*, 453–482.

Polson, N. G., and J. G. Scott (2012), On the half-cauchy prior for a global scale parameter, *Bayesian Analysis*, *7*(4), 887–902.

Pope, S. B. (2000), *Turbulent Flows*, Cambridge University Press, doi:10.1017/CBO9780511840531.

Pope, S. B. (2001), Turbulent flows.

Proctor, J. L., S. L. Brunton, and J. N. Kutz (2016), Dynamic mode decomposition with control, *SIAM Journal on Applied Dynamical Systems*, *15*(1), 142–161.

Proctor, J. L., S. L. Brunton, and J. N. Kutz (2018), Generalizing koopman theory to allow for inputs and control, *SIAM Journal on Applied Dynamical Systems*, *17*(1), 909–930.

Rahimi, A., and B. Recht (2008), Random features for large-scale kernel machines, in *Advances in neural information processing systems*, pp. 1177–1184.

Ramachandran, P., B. Zoph, and Q. V. Le (2017), Searching for activation functions, *arXiv preprint arXiv:1710.05941*.

Raman, V., H. Pitsch, and R. O. Fox (2005), Hybrid large-eddy simulation/lagrangian filtered-density-function approach for simulating turbulent combustion, *Combustion and Flame*, *143*(1-2), 56–78.

Rasmussen, C. E. (2003), Gaussian processes in machine learning, in *Summer School on Machine Learning*, pp. 63–71, Springer.

Ren, S., K. He, R. Girshick, and J. Sun (2015), Faster r-cnn: Towards real-time object detection with region proposal networks, in *Advances in neural information processing systems*, pp. 91–99.

Ritter, H., A. Botev, and D. Barber (2018), A scalable laplace approximation for neural networks, in *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, vol. 6, International Conference on Representation Learning.

Röjsel, J. (2017), Koopman mode analysis of the side-by-side cylinder wake, *Tech. rep.*, Royal Institute of Technology, Department of Mechanics.

Ross, S. M., et al. (1996), *Stochastic processes*, vol. 2, Wiley New York.

Rowley, C. W. (2005), Model reduction for fluids, using balanced proper orthogonal decomposition, *International Journal of Bifurcation and Chaos*, *15*(03), 997–1013.

Rowley, C. W., and S. T. Dawson (2017), Model reduction for flow analysis and control, *Annual Review of Fluid Mechanics*, *49*, 387–417.

Rowley, C. W., I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson (2009), Spectral analysis of nonlinear flows, *Journal of fluid mechanics*, *641*, 115–127.

Rudy, S. H., J. N. Kutz, and S. L. Brunton (2019), Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, *Journal of Computational Physics*, *396*, 483–506.

Sauer, T., and J. A. Yorke (1993), How many delay coordinates do you need?, *International Journal of Bifurcation and Chaos*, *3*(03), 737–744.

Sauer, T., J. A. Yorke, and M. Casdagli (1991), Embedology, *Journal of statistical Physics*, *65*(3-4), 579–616.

Sayadi, T., J. Nichols, P. Schmid, and M. Jovanovic (2012), Dynamic mode decomposition of h-type transition to turbulence, in *Proceedings of the Summer Program*, pp. 5–14, Citeseer.

Sayadi, T., P. J. Schmid, J. W. Nichols, and P. Moin (2014), Reduced-order representation of near-wall structures in the late transitional boundary layer, *Journal of fluid mechanics*, *748*, 278–301.

Scarano, F. (2012), Tomographic piv: principles and practice, *Measurement Science and Technology*, *24*(1), 012,001.

Schilder, F., W. Vogt, S. Schreiber, and H. M. Osinga (2006), Fourier methods for quasi-periodic oscillations, *International journal for numerical methods in engineering*, *67*(5), 629–671.

Schmid, P. J. (2007), Nonmodal stability theory, *Annu. Rev. Fluid Mech.*, *39*, 129–162.

Schmid, P. J. (2010), Dynamic mode decomposition of numerical and experimental data, *Journal of fluid mechanics*, *656*, 5–28.

Schmid, P. J., L. Li, M. Juniper, and O. Pust (2011), Applications of the dynamic mode decomposition, *Theoretical and Computational Fluid Dynamics*, *25*(1-4), 249–259.

Schmid, P. J., D. Violato, and F. Scarano (2012), Decomposition of time-resolved tomographic piv, *Experiments in Fluids*, *52*(6), 1567–1579.

Sharma, A., J. Xu, A. K. Padthe, P. P. Friedmann, and K. Duraisamy (2019), Simulation of maritime helicopter dynamics during approach to landing with time-accurate wind-over-deck, in *AIAA Scitech 2019 Forum*, p. 0861.

Shi, J., S. Sun, and J. Zhu (2018), Kernel implicit variational inference, *international conference on learning representations*.

Song, G., F. Alizard, J.-C. Robinet, and X. Gloerfelt (2013), Global and koopman modes analysis of sound generation in mixing layers, *Physics of Fluids*, *25*(12), 124,101.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014), Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, *15*(1), 1929–1958.

Stark, J., D. S. Broomhead, M. Davies, and J. Huke (2003a), Delay embeddings for forced systems. II. stochastic forcing, *Journal of Nonlinear Science*, *13*(6), 519–577.

Stark, J., D. S. Broomhead, M. E. Davies, and J. Huke (2003b), Delay embeddings for forced systems. I. Deterministic forcing, *Journal of Nonlinear Science*, *13*(6), 519–577, doi:10.1007/s00332-003-0534-4.

Sugihara, G., B. T. Grenfell, and R. M. May (1990), Distinguishing error from chaos in ecological time series, *Phil. Trans. R. Soc. Lond. B*, *330*(1257), 235–251.

Surana, A., and A. Banaszuk (2016), Linear observer synthesis for nonlinear systems using koopman operator framework, *IFAC-PapersOnLine*, *49*(18), 716–723.

Swischuk, R., B. Kramer, C. Huang, and K. Willcox (2020), Learning physics-based reduced-order models for a single-injector combustion process, *AIAA Journal*, pp. 1–15.

Taira, K., M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. Dawson, and C.-A. Yeh (2019), Modal analysis of fluid flows: Applications and outlook, *AIAA Journal*, pp. 1–25.

Taira, K., M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. Dawson, and C.-A. Yeh (2020), Modal analysis of fluid flows: Applications and outlook, *AIAA Journal*, *58*(3), 998–1022.

Taira, K., et al. (2017), Modal analysis of fluid flows: An overview, *Aiaa Journal*, pp. 4013–4041.

Takeishi, N., Y. Kawahara, and T. Yairi (2017), Learning koopman invariant subspaces for dynamic mode decomposition, in *Advances in Neural Information Processing Systems*, pp. 1130–1140.

Takens, F. (1981), Detecting strange attractors in turbulence, in *Dynamical systems and turbulence, Warwick 1980*, pp. 366–381, Springer.

Tissot, G., L. Cordier, N. Benard, and B. R. Noack (2014), Model reduction using dynamic mode decomposition, *Comptes Rendus Mécanique*, *342*(6-7), 410–416.

Towne, A., O. T. Schmidt, and T. Colonius (2018), Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis, *Journal of Fluid Mechanics*, *847*, 821–867.

Tran, D., A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei (2016), Edward: A library for probabilistic modeling, inference, and criticism, *arXiv preprint arXiv:1610.09787*.

Trefethen, L. N., A. E. Trefethen, S. C. Reddy, and T. A. Driscoll (1993), Hydrodynamic stability without eigenvalues, *Science*, *261*(5121), 578–584.

Tu, J. H., C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz (2013), On dynamic mode decomposition: theory and applications, *arXiv preprint arXiv:1312.0041*.

Vapnik, V. (1992), Principles of risk minimization for learning theory, in *Advances in neural information processing systems*, pp. 831–838.

Vapnik, V. N., and A. Y. Chervonenkis (2015), On the uniform convergence of relative frequencies of events to their probabilities, in *Measures of complexity*, pp. 11–30, Springer.

Vautard, R., P. Yiou, and M. Ghil (1992), Singular-spectrum analysis: A toolkit for short, noisy chaotic signals, *Physica D: Nonlinear Phenomena*, *58*(1-4), 95–126.

Verzicco, R., and R. Camussi (2003), Numerical experiments on strongly turbulent thermal convection in a slender cylindrical cell, *Journal of Fluid Mechanics*, *477*, 19–49.

Wang, L., J. Wang, J. Xiang, and H. hui Yue (2019), A re-weighted smoothed l 0-norm regularized sparse reconstructed algorithm for linear inverse problems, *Journal of Physics Communications*.

Wehmeyer, C., and F. Noé (2018), Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics, *The Journal of Chemical Physics*, *148*(24), 241,703.

White, F. M., and I. Corfield (2006), *Viscous fluid flow*, vol. 3, McGraw-Hill New York.

Williams, M. O., C. W. Rowley, and I. G. Kevrekidis (2014), A kernel-based approach to data-driven koopman spectral analysis, *arXiv preprint arXiv:1411.2260*.

Williams, M. O., I. G. Kevrekidis, and C. W. Rowley (2015), A data–driven approximation of the koopman operator: Extending dynamic mode decomposition, *Journal of Nonlinear Science*, *25*(6), 1307–1346.

Xu, B., R. Huang, and M. Li (2016), Revise saturated activation functions, *arXiv preprint arXiv:1602.05980*.

Xu, J., and K. Duraisamy (2019), Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics, *arXiv preprint arXiv:1912.11114*.

Xu, J., C. Huang, and K. Duraisamy (2020), Reduced-order modeling framework for combustor instabilities using truncated domain training, *AIAA Journal*, *58*(2), 618–632.

Yeung, E., S. Kundu, and N. Hodas (2017), Learning deep neural network representations for koopman operators of nonlinear dynamical systems, *arXiv preprint arXiv:1708.06850*.

Yeung, E., S. Kundu, and N. Hodas (2019), Learning deep neural network representations for koopman operators of nonlinear dynamical systems, in *2019 American Control Conference (ACC)*, pp. 4832–4839, IEEE.

Yu, H. (2012), jet-flame stabilization of ethylene.

Yuan, W., A. Wall, and R. Lee (2018), Combined numerical and experimental simulations of unsteady ship airwakes, *Computers & Fluids*, *172*, 29–53.

Zhang, C., J. Butepage, H. Kjellstrom, and S. Mandt (2018), Advances in variational inference, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, doi:10.1109/TPAMI.2018.2889774.

Zhang, H., S. Dawson, C. W. Rowley, E. A. Deem, and L. N. Cattafesta (2017), Evaluating the accuracy of the dynamic mode decomposition, *arXiv preprint arXiv:1710.00745*.

Zhang, L., and H. Schaeffer (2019), On the convergence of the sindy algorithm, *Multiscale Modeling & Simulation*, *17*(3), 948–972.

Zhao, L., Q. Sun, J. Ye, F. Chen, C.-T. Lu, and N. Ramakrishnan (2015), Multi-task learning for spatio-temporal event forecasting, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1503–1512, ACM.

Zhu, Y., and N. Zabaras (2018), Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification, *Journal of Computational Physics*, *366*, 415–447.

Zhu, Y., N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris (2019), Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *arXiv preprint arXiv:1901.06314*.

Zhuo, J., C. Liu, J. Shi, J. Zhu, N. Chen, and B. Zhang (2017), Message Passing Stein Variational Gradient Descent, *arXiv e-prints*, arXiv:1711.04425.

Zou, H., and T. Hastie (2005), Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.