Learning the Koopman Eigendecomposition: A Diffeomorphic Approach

Petar Bevanda, Johannes Kirmayr, Stefan Sosnowski, Sandra Hirche

***Abstract*— We present a novel data-driven approach for learning linear representations of a class of stable nonlinear systems using Koopman eigenfunctions. Utilizing the spectral equivalence of topologically conjugate systems, we construct Koopman eigenfunctions corresponding to the nonlinear system to form linear predictors of nonlinear systems. The conjugacy map between a nonlinear system and its Jacobian linearization is learned via a diffeomorphic neural network. The latter allows for a well-defined, supervised learning problem formulation. Given the learner is diffeomorphic per construction, our learned model is asymptotically stable regardless of the representation accuracy. The universality of the diffeomorphic learner leads to the universal approximation ability for Koopman eigenfunctions - admitting suitable expressivity. The efficacy of our approach is demonstrated in simulations.**

I. INTRODUCTION

For complex nonlinear systems, models based on first-principles often do not fully resemble the true system due to unmodeled phenomena. To better deal with the aforementioned, flexible machine learning techniques are employed (e.g. neural networks or Gaussian processes) for prediction [1], [2], model-based control [3] and analysis [4], [5]. Although classical nonlinear system representations enjoy incredible success, multi-step prediction, analysis and optimization-based control are substantially more challenging than that of their linear analogues. Inspired by the infinite-dimensional but *linear* Koopman operator - named after B.O. Koopman's seminal work [6] - a rise of interest for global linearizations is observed in various research fields. Trading infinite-dimensionality for linearity enables the use of efficient linear techniques for nonlinear systems - leading to more challenging identification but efficacious prediction, analysis and control [7]. The challenge of identification involves "lifting" the original system state to suitable higher-dimensional coordinates that represent the linear Koopman operator (generator) in a finite-dimensional form.

A dominant train of thought assumes a predefined library of functions approximating the operator - akin to the well-known extended DMD (EDMD) [8]. However, a "good" library of functions representing the Koopman operator (generator) should be both dynamically closed and relevant for reconstructing the original state evolution. Hence, *apriori* access to a suitable function library is a strong assumption - commonly leading to only locally accurate models and no practicable learning guarantees. Other approaches leverage the expressive power of neural networks or kernel methods to learn a suitable library of functions [9], [10] but often lack theoretical justification.

However, the (generalized) eigenfunctions of the operator are dynamically closed coordinates by definition. Thus, to learn a Koopman operator (generator) representation for long-term accurate prediction, it is vital to construct genuine eigenfunction-coordinates. Nevertheless, very few works consider direct learning of genuine Koopman eigenfunctions for linear prediction models, such as Korda et al. [11]. Although assuming no spectral or feature knowledge, its performance is dependent on trajectory data, a specific choice of eigenfunction lattices and basis functions for interpolation. Similarly to our own, the work of Folkestad et al. [12] proposes to learn a conjugacy between the nonlinear dynamics and its Jacobian linearization to construct Koopman operator eigenfunctions for linear prediction. Crucially, however, the aforementioned method solves a markedly underdetermined learning problem while employing heuristics that are not theoretically justified. Furthermore, although considering stable systems, it provides no stability guarantees that would allow for safety and physical consistency even for unseen states [13].

This paper presents a novel data-driven approach for learning Koopman operator generator eigenfunctions for prediction. To construct Koopman eigenfunctions, we learn a diffeomorphism between a hyperbolic nonlinear system and its linearization using a diffeomorphic neural network. The latter, together with explicit training targets, leads to a well-defined supervised learning problem. The learner's universal approximation capability of diffeomorphisms transfers to that of the nonlinear system's Koopman eigenfunctions - allowing for sufficient expressivity. Additionally, our framework also ensures safety in the sense of guaranteeing global asymptotic stability of the Koopman operator dynamical model - regardless of the representation accuracy. The superior performance of our approach - also compared to existing techniques - is demonstrated in simulation examples.

This paper is structured as follows: After the problem setup and introduction of required system-theoretical results in Sec. II and III, we propose a novel data-driven framework - *KoopmanEigenFlows* - for constructing genuine Koopman operator eigenfunctions in Sec. IV. Thereafter in Sec. V, we propose a linear representation of the nonlinear systems with our algorithm **K**oopman**E**igen**F**low **M**ode **D**ecomposition (KEFMD). It is followed by numerical evaluation in Sec. VI and a conclusion.

This work was supported by European Union's Horizon 2020 research and innovation programme under grant agreement no. 871295 "SeaClear" (SEarch, identificAtion and Collection of marine Litter with Autonomous Robots).

All authors are members of the Chair of Information-oriented Control, Department of Electrical and Computer Engineering, Technical University of Munich, D-80333 Munich, Germany, {petar.bevanda, johannes.kirmayr, sosnowski, hirche}@tum.de.

II. Problem Formulation

Consider a partially known, continuous-time nonlinear system [1]

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{r}(\boldsymbol{x}) \tag{1}$$

with continuous states on a compact set $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^d$ containing the origin, consisting of a known $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ and an unknown $\boldsymbol{r} : \mathbb{X} \mapsto \mathbb{R}^d$.

Assumption 2.1: We assume that the origin of (1) is globally exponentially stable.

The above system class includes dynamical systems representing motion as well as various dissipative Lagrangian systems. Due to their continuous-time nature, the dynamics are fully described by the forward-complete flow map [14] of (1) given by

$$\boldsymbol{x}(t_0) \equiv \boldsymbol{x}_0, \quad \boldsymbol{F}^t(\boldsymbol{x}_0) := \boldsymbol{x}_0 + \int_{t_0}^{t_0+t} \boldsymbol{f}(\boldsymbol{x}(\tau))d\tau, \tag{2}$$

which has a unique solution on $[0, +\infty)$ from the initial condition $\boldsymbol{x}$ at $t = 0$ due to stability of the isolated attractor [15]. This flow map naturally induces the associated Koopman operator semigroup as defined in the following.

Definition 2.1: The semigroup of Koopman operators $\{\mathcal{K}^t\}_{t\in\mathbb{R}_{+,0}} : C(\mathbb{X}) \mapsto C(\mathbb{X})$ for the flow (2) acts on a scalar observable function $h \in C(\mathbb{X})$ on the state space $\mathbb{X}$ through $\mathcal{K}^t_{\boldsymbol{f}} h = h \circ \boldsymbol{F}^t$.

Definition 2.2 ([16]): The operator $\mathcal{G}_{\mathcal{K}}$, is the infinitesimal generator

$$\mathcal{G}_{\mathcal{K}} h = \lim_{t\to 0^+} \frac{\mathcal{K}^t h - h}{t} = \frac{d}{dt} h, \tag{3}$$

of the time-t indexed semigroup of Koopman operators $\{\mathcal{K}^t\}_{t\in\mathbb{R}_{+,0}}$.

The natural linearly evolving coordinates are the eigenfunctions of evolution operators.

Definition 2.3: An observable $\phi \in C(\mathbb{X})$ is an *eigenfunction* if it satisfies

$$[\mathcal{G}_{\mathcal{K}}\phi](\boldsymbol{x}) = \dot{\phi}\,(\boldsymbol{x}) = \lambda\phi(\boldsymbol{x}), \tag{4}$$

associated with the *eigenvalue* $\lambda \in \mathbb{C}$.

Property 2.1: Since $\mathcal{G}_{\mathcal{K}}$ is the infinitesimal generator of the semigroup of Koopman operators $\{\mathcal{K}^t\}_{t\in\mathbb{R}_{+,0}}$, the following is also satisfied

$$[\mathcal{K}^t_{\boldsymbol{f}}\phi](\boldsymbol{x}) = \phi\left(\boldsymbol{F}^t(\boldsymbol{x})\right) = e^{\lambda t}\phi\left(\boldsymbol{x}\right), \tag{5}$$

along the vector field's flow.

Due to Assumption 2.1, the Koopman operator generator has a pure point spectrum for the dynamics (1) [17]. Thus, for each observable $\boldsymbol{h}$, there exists a sequence $\boldsymbol{v}_j(h) \in \mathbb{C}$ of mode weights, such that action of the Koopman generator is represented through the following decomposition

$$\dot{\boldsymbol{h}} = \mathcal{G}_{\mathcal{K}}\boldsymbol{h} = \sum_{j=1}^{\infty} \boldsymbol{v}_j(\boldsymbol{h})\left(\mathcal{G}_{\mathcal{K}}\phi_j\right) = \sum_{j=1}^{\infty} \boldsymbol{v}_j(\boldsymbol{h})\lambda_j\phi_j. \tag{6}$$

Given the existence of the decomposition (6), we are interested in learning a model of the following form

$$\boldsymbol{z}_0 = \boldsymbol{\phi}(\boldsymbol{x}(0)), \tag{7a}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{\Lambda}\boldsymbol{z}, \tag{7b}$$

$$\boldsymbol{x} = \boldsymbol{V}\boldsymbol{z}, \tag{7c}$$

where $\boldsymbol{\phi} = [\phi_1, \cdots, \phi_D]^\top$ are the finite-dimensional eigenfunction coordinates, $\boldsymbol{\Lambda} \in \mathbb{R}^{D\times D}$ and $\boldsymbol{V} \in \mathbb{R}^{d\times D}$. We consider the full-state observable to be the output of interest $\boldsymbol{h}(\boldsymbol{x}) = \mathrm{id}(\boldsymbol{x})$ in (7c). The goal is to trade the nonlinearity of a d-dimensional ODE (1) for a nonlinear "lift" (7a) of the initial condition $\boldsymbol{x}(0)$ to higher dimension ($D \gg d$) leading to a linearly evolving model $\dot{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{\phi}(\boldsymbol{x})$ with a closed form flow $\boldsymbol{x}(t) = \boldsymbol{V}e^{\boldsymbol{\Lambda}t}\boldsymbol{\phi}(\boldsymbol{x}(0))$. In general, the output of interest (7c) can be any other observable function $\boldsymbol{h}(\boldsymbol{x})$ as well.

III. Modeling via Equivalence Relations

To reliably construct Koopman eigenfunctions, we utilize equivalence relations between the nonlinear system (1) and its linearization around the origin. To utilize the aforementioned to build models of the form (7), we introduce some relevant Koopman eigenfunction properties.

Property 3.1 ([18]): If the function space of eigenfunction is chosen to be a Banach algebra (e.g. $C^1(\mathbb{X})$), the set of eigenfunctions forms an Abelian semigroup under point-wise products of functions. Thus, for $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$ with eigenvalues λ_1 and λ_2, $\phi_1\phi_2$ is also an eigenfunction of $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$ with eigenvalue $\lambda_1 + \lambda_2$.

Definition 3.1 ([12], [19]): Consider a system $\dot{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{y}$ and a multi-index $\boldsymbol{m} = [m_1, ..., m_d] \in \mathbb{N}_0^d$ such that $\|\boldsymbol{m}\|_1 = m_1 + \cdots + m_d \leq p$ for $p \in \mathbb{N}$. Consider $\{E^p\}$ to be the eigenpair group - a collection of all eigenvalue-eigenfunction pairs - of the Koopman operator generator $\mathcal{G}_{\mathcal{K}_{\boldsymbol{A}}}$ for $\dot{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{y}$ with its minimal group generator $\mathcal{P}_E$:

$$\{E^p\} = \left\{ \left(\sum_{i=1}^{p} m_i\lambda_i, \prod_{i=1}^{p} \varphi_i^{m_i} \right) \mid (\lambda_i, \varphi_i) \subset \mathcal{P}_E \right\}. \tag{8}$$

Then, the elements of $\mathcal{P}_E$ are *principle* eigenvalues-eigenfunction pairs (λ_i, φ_i) of $\mathcal{G}_{\mathcal{K}_{\boldsymbol{A}}}$.

Less formally, *principle eigenpairs* form the minimal set used to construct arbitrarily many other eigenpairs (8).

A. Topological Proxy to Koopman Eigenfunctions

Here, we define the notions relevant for the geometric equivalence relations considered in this work.

Definition 3.2: Consider a bijective map $\boldsymbol{g} : \mathbb{R}^n \mapsto \mathbb{R}^n$. The bijective map $\boldsymbol{g}$ is a *homeomorphism* if both the map and its inverse $\boldsymbol{g}^{-1}$ are continuous. If the maps $\boldsymbol{g}$ and $\boldsymbol{g}^{-1}$ are also continuously differentiable, then $\boldsymbol{g}$ is a *diffeomorphism.*

Definition 3.3: Two flows $\boldsymbol{F}^t : \mathbb{X} \mapsto \mathbb{X}$ and $\boldsymbol{C}^t : \mathbb{Y} \mapsto \mathbb{Y}$ of vector fields $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ and $\dot{\boldsymbol{y}} = \boldsymbol{c}(\boldsymbol{y})$ are topologically

[1] **Notation:** Lower/upper case bold symbols $\boldsymbol{x}$/$\boldsymbol{X}$ denote vectors/matrices. Symbols $\mathbb{N}/\mathbb{R}/\mathbb{C}$ denote sets of natural/real/complex numbers while $\mathbb{N}_0$ denotes all natural numbers with zero and $\mathbb{R}_{+,0}/\mathbb{R}_+$ all positive reals with/without zero. Function spaces with a specific integrability/smoothness order are denoted as L/C with the order (class) specified in their exponent. The Jacobian matrix of map $\boldsymbol{h}$ evaluated at $\boldsymbol{x}$ is denoted as $\boldsymbol{J}_{\boldsymbol{h}}(\boldsymbol{x})$.

A flow induced by a vector field $\dot{x} = f(x)$ is denoted as $F^t(x)$ with its associated family of composition (Koopman) operators $\{\mathcal{K}^t_f\}_{t\in\mathbb{R}_{+,0}}$. The L^p-norm on a set $\mathbb{X}$ is denoted as $\|\cdot\|_{p,\mathbb{X}}$.

conjugate if there exists a homeomorphism $\boldsymbol{g}:\mathbb{X}\mapsto\mathbb{Y}$ such that $\boldsymbol{g}\circ\boldsymbol{F}^t=\boldsymbol{C}^t\circ\boldsymbol{g}$ holds $\forall\boldsymbol{x}\in\mathbb{X}$ and $t\in\mathbb{R}$.

Proposition 3.1 ([18]): Consider the same two flows from Definition 3.3. If $(e^{\lambda t},\varphi)$ is an eigenpair of $\mathcal{K}_{\boldsymbol{c}}^t$, then $(e^{\lambda t},\varphi\circ\boldsymbol{g})$ is an eigenpair of $\mathcal{K}_{\boldsymbol{f}}^t$.

Extending the topological conjugacy to the entire region of attraction is formalized in the following.

Proposition 3.2 ([12]): Assume that the nonlinear system (1) is topologically conjugate to its Jacobian linearization via the diffeomorphism $\boldsymbol{d}:\mathbb{X}\mapsto\mathbb{Y}$. Let $\mathbb{B}\subset\mathbb{X}$ be a simply connected, bounded, positively invariant open set in $\mathbb{X}$ such that $\boldsymbol{d}(\mathbb{B})\subset\mathbb{Q}_r\subset\mathbb{Y}$, where $\mathbb{Q}_r$ is a cube in $\mathbb{Y}$. Scaling $\mathbb{Q}_r$ to the unit cube $\mathbb{Q}_1$ via the diffeomorphism $\boldsymbol{g}:\mathbb{Q}_r\mapsto\mathbb{Q}_1$ gives $(\boldsymbol{g}\circ\boldsymbol{d})(\mathbb{B})\subset\mathbb{Q}_1$. Then, if φ is an eigenfunction for $\mathcal{K}_{\boldsymbol{A}}^t$ at $e^{\lambda t}$, then $\varphi\circ\boldsymbol{g}\circ\boldsymbol{d}$ is an eigenfunction for $\mathcal{K}_{\boldsymbol{f}}^t$ at eigenvalue $e^{\lambda t}$, where $\mathcal{K}_{\boldsymbol{f}}^t$ is the Koopman operator semigroup associated with the nonlinear dynamics (1).

Theorem 3.1 ([20]): Consider the system (1) with $\boldsymbol{r}(\boldsymbol{x})\in C^2(\mathbb{X})$. Under Assumption 2.1, the matrix $\boldsymbol{A}$ in (1) is Hurwitz i.e., all eigenvalues have negative real parts. Then in the region of attraction $\mathbb{S}$ of the origin there exists $\boldsymbol{\varrho}(\boldsymbol{x})\in C^1(\mathbb{S}):\mathbb{S}\mapsto\mathbb{R}^d$, such that $\boldsymbol{y}=\boldsymbol{d}(\boldsymbol{x})=\boldsymbol{x}+\boldsymbol{\varrho}(\boldsymbol{x})$ is a C^1 diffeomorphism with $\boldsymbol{\varrho}(\boldsymbol{0})=\boldsymbol{0}$ in $\mathbb{S}$ and satisfying $\dot{\boldsymbol{y}}=\boldsymbol{A}\boldsymbol{y}$.

By utilizing topological conjugacy, one exploits the fact that the eigenvalues are shared between the full nonlinear system (1) and its linearization around an asymptotically stable fixed point. Subsequently, one is able to construct arbitrarily many eigenfunctions from the principal ones via Definition 3.1 and Theorem 3.1. Using these properties allows for learning Koopman-based dynamical models by lifting to eigenfunction coordinates by design.

B. Asymptotic Stability Guarantees

By Theorem 3.1 we deal with diffeomorphic mappings, motivating the definition of a stronger equivalence notion.

Definition 3.4 ([21]): Vector fields $\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x})$ and $\dot{\boldsymbol{y}}=\boldsymbol{t}(\boldsymbol{y})$ are said to be diffeomorphic, or smoothly equivalent, if there exists a diffeomorphism $\boldsymbol{d}:\mathbb{R}^d\mapsto\mathbb{R}^d$ such that $\forall\boldsymbol{x}\in\mathbb{R}^d$ $\boldsymbol{t}(\boldsymbol{d}(\boldsymbol{x}))=\boldsymbol{J_d}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$ holds.

To ensure safety, we are interested in transferring the asymptotic stability properties of the linearization to the lifted linear system (7). That is enabled by the following result.

Theorem 3.2: Consider a system (1) satisfying Assumption 2.1, its Jacobian linearization $\dot{\boldsymbol{y}}=\boldsymbol{A}\boldsymbol{y}$ and a diffeomorphic map $\boldsymbol{d}$. Let the Koopman eigenfunctions (7a) corresponding to $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$ of system (1) be constructed via Proposition 3.2 utilizing Definition 3.1. Then, the associated $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$-realization of the form (7) is guaranteed to be asymptotically stable.

Proof: First we show the transition matrix in (7b) is Hurwitz. As $\boldsymbol{A}$ is Hurwitz by Assumption 2.1, then $\boldsymbol{\Lambda}$ in (7b) is as well with eigenvalues satisfying $\mathrm{Re}[\sum_{i=1}^{p}m_i\lambda_i]<0$ per Definition 3.1. Secondly, we show the lifting (7a) is an immersion - $\mathrm{rank}(\boldsymbol{J_\phi}(\boldsymbol{x}))=\mathrm{dim}(\boldsymbol{x})$. Consider an eigenfunction library $\boldsymbol{\varphi}(\boldsymbol{y})$ constructed by concatenating elements of $\{E^p\}$ of Definition 3.1. As it forms a monomial basis, its rank equals $\mathrm{dim}(\boldsymbol{y})$ making it an immersion. Diffeomorphisms $\boldsymbol{g}$ and $\boldsymbol{d}$ prescribed by Proposition 3.2 are immersions by definition. As compositions of immersions are an immersion, the immersibility of $\boldsymbol{\phi}=\boldsymbol{\varphi}\circ\boldsymbol{g}\circ\boldsymbol{d}$ is ensured. With $\boldsymbol{\Lambda}$ Hurwitz and $\boldsymbol{\phi}$ immersible, the asymptotic stability of the lifted model (7) follows via [22, Proposition 1]. ■

Remark 3.1: The result of Theorem 3.2 establishes that the asymptotic stability of the Jacobian linearization carries over to the linear predictor of the form (7) when constructed by diffeomorphically transforming the eigenfunctions of a linear system.

IV. Learning Equivalences via Invertible Neural Networks

In order to learn the Koopman eigenfunctions through an equivalence relation for the system (1) in a well-conditioned manner, one needs to ensure the function approximator is constrained to be a diffeomorphism. Allowing for that are flow-based neural networks, where coupling flow invertible neural networks (CF-INN) present a powerful tool. Although with their form restricted compared to vanilla neural networks, there are CF-INN architectures exhibiting L^p-universality/sup-universality for a large class of diffeomorphisms [23]. Hence, they can be relied on for learning of Koopman eigenfunctions via Proposition 3.2..

Assumption 4.1: Let $D^2(\mathbb{X})$ be the set of all $C^2(\mathbb{X})$ diffeomorphisms. Then, there is a class $\mathcal{D}$ of universal approximators $\hat{\boldsymbol{d}}$ such that for any $\boldsymbol{d}\in D^2(\mathbb{X})$ and any $\varepsilon>0,\exists\hat{\boldsymbol{d}}\in\mathcal{D}$ such that $\|\boldsymbol{d}-\hat{\boldsymbol{d}}\|_{p,\mathbb{X}}<\varepsilon$.

The above assumption is hardly restrictive as it is fulfilled by almost all diffeomorphisms and thus systems (1). Therefore, we can state the following result on the expressivity of learning Koopman eigenfunctions constructed via Proposition 3.2.

Lemma 4.1: Let there exist a C^2-diffeomorphism making the system (1) smoothly equivalent to its Jacobian linearization. Consider a function approximator $\hat{\boldsymbol{d}}$ fulfilling Assumption 4.1 and let $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$-eigenfunctions of (1) be constructed via Proposition 3.2 such that an approximate eigenfunction has the form $\hat{\phi}=\varphi\circ\boldsymbol{g}\circ\hat{\boldsymbol{d}}$. Then, for any $\delta>0$ and any ϕ, $\exists\hat{\phi}$ such that $\|\phi-\hat{\phi}\|_{p,\mathbb{X}}<\delta$.

Proof: Due to continuity on their corresponding domains, there exist Lipschitz constants L_φ and $L_{\boldsymbol{g}}$, associated to φ and $\boldsymbol{g}$ respectively so that $\|\boldsymbol{d}-\hat{\boldsymbol{d}}\|_{p,\mathbb{X}}<\frac{\delta}{L_\varphi L_{\boldsymbol{g}}}$ holds. Then, we have

$$\begin{aligned}\|\phi-\hat{\phi}\|_{p,\mathbb{X}}&\overset{\text{Prop. 3.2}}{=}\|\varphi\circ\boldsymbol{g}\circ\boldsymbol{d}-\varphi\circ\boldsymbol{g}\circ\hat{\boldsymbol{d}}\|_{p,\mathbb{X}}\\&\leq\quad L_\varphi L_{\boldsymbol{g}}\|\boldsymbol{d}-\hat{\boldsymbol{d}}\|_{p,\mathbb{X}}<\delta,\end{aligned}\tag{9}$$

concluding the proof. ■

A. Affine Coupling Flows (ACF)

The approach of NF is to compose a complicated bijective function successively from multiple simpler bijections - using the fact that the composition of bijective functions is again bijective. In our case, we form a diffeomorphism between the nonlinear system and its linearization by multiple simpler diffeomorphisms $\hat{\boldsymbol{d}}_{\boldsymbol{i}}$ so that $\boldsymbol{y}=\hat{\boldsymbol{d}}(\boldsymbol{x})=\hat{\boldsymbol{d}}_{\boldsymbol{k}}\circ...\circ\hat{\boldsymbol{d}}_{\boldsymbol{1}}(\boldsymbol{x})$ - as visualized in Fig. 1. The bijectivity of the individual

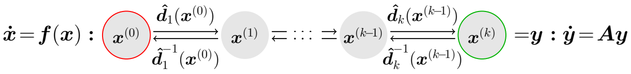
$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x})$:
$\boldsymbol{x}^{(0)}$
$\hat{\boldsymbol{d}}_1(\boldsymbol{x}^{(0)})$
$\hat{\boldsymbol{d}}_1^{-1}(\boldsymbol{x}^{(0)})$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(k-1)}$
$\hat{\boldsymbol{d}}_k(\boldsymbol{x}^{(k-1)})$
$\hat{\boldsymbol{d}}_k^{-1}(\boldsymbol{x}^{(k-1)})$
$\boldsymbol{x}^{(k)}$
$=\boldsymbol{y} : \dot{\boldsymbol{y}}=\boldsymbol{A}\boldsymbol{y}$

Fig. 1: Construction of a linearizing diffeomorphism

functions $\hat{\boldsymbol{d}}_{\boldsymbol{i}}$ is ensured by a special structure - called affine coupling layers

$$\hat{\boldsymbol{d}}_i(\boldsymbol{x}^{(i)}) = \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{a}}^{(i)} \\ \boldsymbol{x}_{\boldsymbol{b}}^{(i)} \odot \exp(\boldsymbol{s}_{\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{a}}^{(i)})) + \boldsymbol{t}_{\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{a}}^{(i)}) \end{bmatrix}, \tag{10}$$

where $\boldsymbol{x}_{\boldsymbol{a}} \equiv x_1, ..., x_n$, $\boldsymbol{x}_{\boldsymbol{b}} \equiv x_{n+1}, ..., x_N$, $\odot$ denotes Hadamard product and exp denotes pointwise exponential. The input vector $\boldsymbol{x}^{(i)}$ is split dimension-wise into two parts $\boldsymbol{x}_a^{(i)}$ and $\boldsymbol{x}_b^{(i)}$; $\boldsymbol{x}_a^{(i)}$ is then scaled with $\exp(\boldsymbol{s}(\boldsymbol{x}))$, and translated with $\boldsymbol{t}(\boldsymbol{x})$, and multiplied/added element-wise to $\boldsymbol{x}_b^{(i)}$. It is important to change the policy of splitting for each affine coupling layer to leave any component unaltered. The scaling functions $\boldsymbol{s}_i : \mathbb{R}^n \mapsto \mathbb{R}^{N-n}$ and translation functions $\boldsymbol{t}_i : \mathbb{R}^n \mapsto \mathbb{R}^{N-n}$ are learned and parametrized by multilayer neural networks with suitably smooth activation functions to form a diffeomorphism. The parameters that have to be trained in order to learn a diffeomorphisms are therefore the weights and biases in the neural networks of the scaling and translation functions - concatenated in parameters $\boldsymbol{w} = [\boldsymbol{w}_{\boldsymbol{s}_1}^\top, \boldsymbol{w}_{\boldsymbol{t}_1}^\top, \cdots, \boldsymbol{w}_{\boldsymbol{s}_k}^\top, \boldsymbol{w}_{\boldsymbol{t}_k}^\top]^\top$.

Remark 4.1: As the special structure of the affine coupling layer already ensures bijectivity and invertibility by design, there are no restrictions to the neural networks of the scaling and translation functions and they can be learned freely.

B. Supervised Learning of a Linearizing Diffeomorphism

As a corollary of Theorem 3.1 and Definition 3.4 the following equations

$$\dot{\boldsymbol{x}}=\boldsymbol{J}_{\boldsymbol{d}}^{-1}(\boldsymbol{x})\boldsymbol{A}\boldsymbol{d}(\boldsymbol{x}),\ \boldsymbol{J}_{\boldsymbol{d}}(\boldsymbol{0})=\boldsymbol{I},\ \boldsymbol{d}(\boldsymbol{0})=\boldsymbol{0}, \tag{11}$$

are to be fulfilled by a linearizing diffeomorphism. Assume the availability of a data-set of N input-output pairs $\mathbb{D}_N = \left\{\dot{\boldsymbol{x}}^{(i)}, \boldsymbol{x}^{(i)}\right\}_{i=1}^N$ for the system (1) satisfying Assumption 2.1. Then, the solution $\hat{\boldsymbol{d}}(\boldsymbol{x}) := \boldsymbol{d}_{\hat{\boldsymbol{w}}}(\boldsymbol{x})$ to (11) can be obtained in terms of the ACF parameters $\boldsymbol{w}$ by solving the following optimization problem

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \sum_{i=1}^N \|\dot{\boldsymbol{x}} - \boldsymbol{J}_{\boldsymbol{d}_{\boldsymbol{w}}}^{-1}(\boldsymbol{x})\boldsymbol{A}\boldsymbol{d}_{\boldsymbol{w}}(\boldsymbol{x})\|_2^2 + \tag{12a}$$

$$\|\boldsymbol{J}_{\boldsymbol{d}}(\boldsymbol{0}) - \boldsymbol{I}\|_2^2 + \|\boldsymbol{d}(\boldsymbol{0}) - \boldsymbol{0}\|_2^2. \tag{12b}$$

C. Constructing Nonlinear System's Eigenfunctions

Algorithm 1 provides a pseudo code for the construction of Koopman eigenfunctions. Since the eigenfunctions are constructed via the learned diffeomorphism through Normalizing Flows, we call this approach *KoopmanEigenFlows*. We first calculate the eigenfunctions of the linearized system. Let $\boldsymbol{v} = \{\boldsymbol{v}_{\boldsymbol{1}}, ..., \boldsymbol{v}_{\boldsymbol{d}}\}$ be the eigenvector-basis of matrix $\boldsymbol{A}$ corresponding to non-zero eigenvalues $\{\lambda_1, ..., \lambda_d\}$. Then the adjoint basis $\boldsymbol{w} = \{\boldsymbol{w}_{\boldsymbol{1}}, ..., \boldsymbol{w}_{\boldsymbol{d}}\}$ is given via the transposed cofactor-matrix of $\boldsymbol{v}$ so that $\langle \boldsymbol{v}_{\boldsymbol{i}}, \boldsymbol{w}_{\boldsymbol{j}} \rangle = \delta_{ij}$ and $\boldsymbol{w}_{\boldsymbol{k}}$ is an eigenvector of $\boldsymbol{A}^*$ at eigenvalue $\bar{\lambda}_k$. Then, the inner product $\varphi_{\mathrm{p},i} = \langle \boldsymbol{y}, \boldsymbol{w}_{\boldsymbol{j}} \rangle$ is a nonzero principal eigenfunction of the Koopman generator $\mathcal{G}_{\mathcal{K}_{\boldsymbol{A}}}$ of the linearized system cf. [12, Prop. 1]. As a corollary of (6) for observable $\boldsymbol{h} = \mathrm{id}$, we know that the modal decomposition of $\boldsymbol{x}$ is given through

$$\boldsymbol{x} = \sum_{j=1}^{\infty} \boldsymbol{v}_j \phi_j(\boldsymbol{x}) \ , \tag{13}$$

where the infinite sum results from the infinite dimensionality of the Koopman operator. For a practicable representation of (13), we create a library of eigenfunctions taking the principle ones to predefined maximum powers $p^{(1)}, ..., p^{(d)}$ for each principal eigenfunction $\varphi_{\mathrm{p},i}$ using Property 3.1. With this finite number of eigenfunctions, (13) can be written in matrix-vector notation, i.e. $\hat{\boldsymbol{x}} = \boldsymbol{V}\hat{\boldsymbol{\phi}}(\boldsymbol{x}) = \boldsymbol{V}\boldsymbol{z}$, with the reconstruction matrix $\boldsymbol{V}$. The simple library of eigenfunctions of the linearized system $\boldsymbol{\varphi}$ is then converted to a library of eigenfunctions of the nonlinear system via $\hat{\boldsymbol{\phi}}(\boldsymbol{x}) = \boldsymbol{\varphi} \circ \boldsymbol{g} \circ \hat{\boldsymbol{d}}(\boldsymbol{x})$. The eigenvalues are preserved, since they are shared between topologically conjugate systems.

Algorithm 1 KoopmanEigenFlows

```
Input: Jacobian linearization $\boldsymbol{A}$; $\mathbb{D}_N=\{\dot{\boldsymbol{x}}^{(i)}, \boldsymbol{x}^{(i)}\}_{i=1}^N$; maximum powers $p^{(1)}, ..., p^{(d)}$ of the $d$ principal eigenpairs
1: Learn a diffeomorphism $\hat{\boldsymbol{d}}$ through NF:
   $\hat{\boldsymbol{d}}(\boldsymbol{x}) \leftarrow NF(\mathbf{id}(\boldsymbol{x}))$
   $\hat{\boldsymbol{d}}(\boldsymbol{x}) \leftarrow$ (12)
2: Construct principal eigenpairs of the linearized system:
   $\lambda_{\mathrm{p},j}, \varphi_{\mathrm{p},j} \leftarrow \boldsymbol{A}, \langle y, w_j \rangle, \quad j = 1, ..., d$
3: Construct library of eigenpairs of the linearized system:
   for $M$ = all combinations of $[p^{(1)}, ..., p^{(d)}]$ do
       $\lambda_i \leftarrow \sum_{j=1}^d \boldsymbol{M}[i,j] \cdot \lambda_{\mathrm{p},j}, \quad i = 1, ..., \prod_{k=1}^d p^{(k)} + 1$
       $\varphi_i \leftarrow \prod_{j=1}^d \varphi_{\mathrm{p},j}^{\boldsymbol{M}[i,j]}$
4: Construct eigenpairs of the nonlinear system:
   $\hat{\boldsymbol{\phi}} \leftarrow \boldsymbol{\varphi}\left(\boldsymbol{g}(\hat{\boldsymbol{d}}(\boldsymbol{x}))\right), \qquad \boldsymbol{\varphi} = [\varphi_1, ..., \varphi_i, ..., \varphi_D]^T$
Output: A library of eigenvalue-eigenfunction pairs $(\boldsymbol{\lambda}, \hat{\boldsymbol{\phi}})$
```

Corollary 4.1: Let diffeomorphisms $\boldsymbol{d} = \hat{\boldsymbol{d}}_{\boldsymbol{k}} \circ \ldots \circ \hat{\boldsymbol{d}}_{\boldsymbol{1}}(\boldsymbol{x})$ be parameterized through coupling layers (10), which are defined using C^1-functions $\boldsymbol{s}_i$, $\boldsymbol{t}_i$. Then, every solution of the optimization problem (12) constructed via Proposition 3.2 utilizing Definition 3.1 yields a stable system (7).

Proof: As ACFs form diffeomorphisms per construction with C^1-function approximators $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$, the asymptotic stability asserted by Theorem 3.2 holds regardless of the approximation accuracy. ■

Remark 4.2: In essence, the result of Cor. 4.1 decouples safety from performance in Alg. 1 as asymptotic stability is guaranteed regardless of how well (12) fits (11).

V. KoopmanEigenFlow Mode Decomposition (KEFMD)

To construct linear predictors using Koopman eigenfunctions for nonlinear dynamics, we develop a method to build a linear model in the space of Koopman eigenfunction-observables. As *KoopmanEigenFlows* are utilized to construct the Koopman eigenfunction coordinates, we name our algorithm **K**oopman**E**igen**F**low **M**ode **D**ecomposition (KEFMD). Since we inherit the spectrum from the linearization we do not append "Dynamic Mode Decomposition". Furthermore, mode decomposition is a general concept resulting from operator theoretic identification - transcending the original DMD algorithm [24] that only considers observables that are linear functions of the state.

A. The KEFMD Framework

The developments of Sec. III and IV allow for a system identification approach that extends the spectral properties of a Jacobian linearization of the system around an equilibrium to the corresponding nonlinear system. With the output of Algorithm 1, an LTI-system in eigenfunction coordinates - as defined in (7) - is constructed. Once the initial condition is lifted (7a), the evolution (7b) as well as the state reconstruction (7c) is linear - permitted through Assumption 2.1. The corresponding flow of the original state is given explicitly as $\hat{\boldsymbol{x}}(t) = \boldsymbol{V}e^{\boldsymbol{\Lambda}t}\boldsymbol{z}_0$. We can define a discretized matrix $\boldsymbol{\Lambda}_d = e^{\boldsymbol{\Lambda}t}$ by fixing t to a certain time-step value. This way, we obtain a discretized linear evolving model where $\hat{\boldsymbol{x}}^{k+1} = \boldsymbol{V}\boldsymbol{\Lambda}_d^k\boldsymbol{z}_0$. with the discrete-time index k, so that the state $\boldsymbol{x}$ can be predicted linearly. The pseudocode is provided in Algorithm 2.

Algorithm 2 KoopmanEigenFlow Mode Decomposition–KEFMD

```
    Input: Eigenvalue-eigenfunction pairs $(\boldsymbol{\lambda}, \hat{\boldsymbol{\phi}})$ from Alg. 1
1:  Construct the lifted LTI-system:
    $\boldsymbol{\Lambda} \leftarrow \mathrm{diag}(\boldsymbol{\lambda})$
    $\boldsymbol{z} \leftarrow \hat{\boldsymbol{\phi}}(\boldsymbol{x})$
    $\boldsymbol{V} \leftarrow \boldsymbol{x}\boldsymbol{z}^{\dagger}$
2:  $\dot{\hat{\boldsymbol{x}}} \leftarrow \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{z}$
3:  $\hat{\boldsymbol{x}}^{k+1} \leftarrow \boldsymbol{V}\boldsymbol{\Lambda}_d^k\boldsymbol{z}_0$
    Output: $\boldsymbol{\Lambda}, \boldsymbol{\Lambda}_d, \boldsymbol{V}, \hat{\boldsymbol{\phi}}$
```

VI. Evaluation

To validate the proposed approach, we demonstrate the performance of the constructed Koopman operator dynamical models on two examples with different properties. In both, ACF with 7 coupling layers are used, whose composition results in the diffeomorphism. The neural networks for the scaling and translation functions in each affine coupling layers have 3 hidden layers, 120 neurons each with an Exponential Linear Unit (ELU) as the activation function. Batch learning is performed with a batch size of 64 and the initialization of the weights is defined such that $\hat{\boldsymbol{d}}(\boldsymbol{x})$ is an identity map.

Example 6.1 (A simple illustrative example): Consider the following dynamical system [12]:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \mu x_1 \\ \lambda\left(x_2 - x_1^2\right) \end{bmatrix} = \begin{bmatrix} \mu & 0 \\ 0 & \lambda \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \lambda\left(-x_1^2\right) \end{bmatrix}, \quad (14)$$

with $\mu{=}-0.7$ and $\lambda{=}-0.3$. $N{=}4800$ training data is generated from 24 equally long trajectories with sampling time $dt{=}0.065s$, with the starting points are uniformly distributed on the edges of $\mathbb{X}{=}[-5,5]^2$. The maximum powers are set to $p^{(i)}{=}5$ which results in 36 lifted coordinates $\boldsymbol{z}$. For this system the exact diffeomorphism $\boldsymbol{d}(\boldsymbol{x}){=}[x_1, x_2{-}\frac{\lambda}{\lambda-2\mu}x_1^2]^\top$ is known, allowing us to directly compare it to the learned one. Fig. 2 shows the error between the learned and the true diffeomorphism; demonstrating the mapping is well captured by our ACF design from Subs. IV-A.

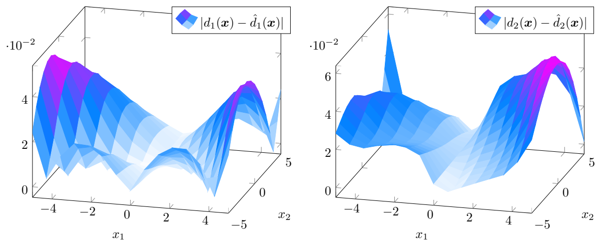

Fig. 2: Dimension-wise plot of the error between the true and learned diffeomorphism.

The main goal, however, is the linear prediction of the state $\boldsymbol{x}$ - given through $\hat{\boldsymbol{x}}^{k+1}{=}\boldsymbol{V}\boldsymbol{\Lambda}_d^k\boldsymbol{z}_0$ in *KEFMD*. To validate our algorithm we compare it with the related KEEDMD [12] and the established EDMD approaches with monomials and radial basis functions - in their continuous-time, Koopman generator versions [25]. In contrast to our deep learning approach, the EDMD approaches use a predefined, function basis as lifting functions. A basis that is not data-driven provides good predictive performance only when it spans a Koopman-invariant subspace. As Koopman-invariant coordinates are unsupervised features - presuming a suitable basis is available *apriori* is a strong assumption. The KEEDMD [12] approach has similar system-theoretic considerations, but employs a very different learning procedure.

Tab. I shows an statistical evaluation (mean and standard deviation) of the root mean squared errors (RMSE's) of 100 trajectories with start points on an uniformly spaced grid for each of the methods. The suitable size of the lifting dimension can be determined via cross-validation. Besides KEFMD, the EDMD-monomials approach gives an almost perfect result since the relevant eigenfunctions for the system of Ex. 6.1 are spanned by monomials.

Example 6.2 (Exact modes unknown apriori): Consider the following dynamical system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \left(a + c \cdot \sin^2(x_2)\right) x_1 \\ b x_2 \end{bmatrix}, \quad (15)$$

with $a{=}-1.3$, $b{=}-2$, $c{=}1.5$ and the Jacobian linearization $\dot{\boldsymbol{y}}{=}[ay_1 \quad by_2]^\top$. The data set with $N{=}11200$ data points is generated from 56 equally long trajectories with sampling time $dt{=}0.015s$, starting on the edges of $\mathbb{X}{=}[-5.5, 5.5]^2$. The number of lifted coordinates is 196. For this system we do not know the explicit diffeomorphism and do not *apriori* know whether an exact finite representation exists.

Fig. 3 shows how our learned Koopman operator dynamical model, although linearly evolving, reproduces a clearly nonlinear time response in original coordinates. Therein are also the resulting state trajectories generated by the different approaches with identical training data. Only KEFMD captures the nonlinear system's behavior well - as quantified in Tab. I and illustrated in Fig. 3. Note that the differing dimension of EDMD-monomials in Ex. 6.2 is due to higher order monomials not performing as well.

Remark 6.1 (Ill-posedness of KEEDMD): The related approach of KEEDMD [12] does not pose a fully supervised learning problem due to not having explicit training targets as our approach does in (12). The aforementioned, coupled with the use of vanilla NNs - a non-diffeomorphic hypothesis class - results in a severely underdetermined formulation. With the addition of employing heuristics that are not theoretically justified, the KEEDMD learning framework is ill-posed - which the performance evaluation in Tab. I demonstrates as well. Even though considering stable systems, it offers no stability guarantees - making the ill-posedness also apparent in Fig. 3 - as the trajectories show no convergence to the origin.

	KEFMD	KEEDMD [2]	EDMD-mon.[3]	EDMD-RBF [2]
6.1	$\mathbf{0.002} \pm 0.001^{(36)}$	$9.12 \pm 6.92^{(36)}$	$\mathbf{0.001} \pm 0.001^{(36)}$	$1.19 \pm 0.52^{(36)}$
6.2	$\mathbf{0.014} \pm 0.000^{(196)}$	$1.41 \pm 0.62^{(196)}$	$0.71 \pm 0.8^{(81)}$	$0.76 \pm 0.32^{(196)}$

TABLE I: RMSE mean and standard deviation for different approaches. The lifted state dimension is written in brackets.

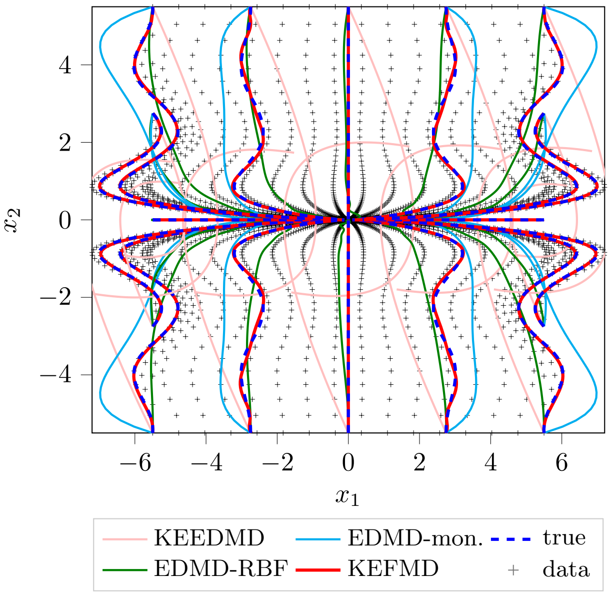

Fig. 3: Linear prediction performance of the different approaches.[4]

VII. Conclusion

In this paper, we present a novel reliable and safe framework for learning Koopman eigenfunctions for constructing linear prediction models for a class of nonlinear dynamics. These results demonstrate superior performance compared to related works and showcase the utility and transferrability of Koopman operator theory to data-driven realizations. The reliable learning of our approach offers extensions to controlled systems for efficient optimal control using linear systems theory.

[2] https://github.com/Cafolkes/keedmd

[3] https://github.com/sklus/d3s

[4] For clarity of presentation every 5th datapoint is plotted.

References

[1] O. Nelles, *Neural Networks.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 239–297.

[2] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith, "Dynamic systems identification with Gaussian processes," vol. 11, no. 4, pp. 411–424, 2005.

[3] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using Gaussian processes," in *2017 IEEE 56th Conference on Decision and Control*, 2017, pp. 5249–5255.

[4] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *2016 IEEE 55th Conference on Decision and Control*, 2016, pp. 4661–4666.

[5] A. Lederer and S. Hirche, "Local asymptotic stability analysis and region of attraction estimation with gaussian processes," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 1766–1771.

[6] B. O. Koopman, "Hamiltonian Systems and Transformation in Hilbert Space," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 17, no. 5, pp. 315–318, 1931.

[7] P. Bevanda, S. Sosnowski, and S. Hirche, "Koopman operator dynamical models: Learning, analysis and control," *Annual Reviews in Control*, vol. 52, pp. 197–212, 2021.

[8] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *Journal of Nonlinear Science volume*, vol. 25, no. 6, pp. 1307–1346, 2015.

[9] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," *Chaos*, vol. 27, no. 10, 2017.

[10] Y. Lian and C. N. Jones, "Learning Feature Maps of the Koopman Operator: A Subspace Viewpoint," in *2019 IEEE 58th Conference on Decision and Control*, 2019, pp. 860–866.

[11] M. Korda and I. Mezić, "Optimal construction of Koopman eigenfunctions for prediction and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5114–5129, 2020.

[12] C. Folkestad, D. Pastor, I. Mezić, R. Mohr, M. Fonoberova, and J. Burdick, "Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control," in *2020 American Control Conference (ACC)*, 2020b, pp. 3906–3913.

[13] S. Tesfazgi, A. Lederer, and S. Hirche, "Inverse reinforcement learning: A control lyapunov approach," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 3627–3632.

[14] A. Bittracher, P. Koltai, and O. Junge, "Pseudogenerators of spatial transfer operators," *SIAM Journal on Applied Dynamical Systems*, vol. 14, no. 3, pp. 1478–1517, 2015.

[15] D. Angeli and E. D. Sontag, "Forward completeness, unboundedness observability, and their Lyapunov characterizations," *Systems and Control Letters*, vol. 38, no. 4-5, pp. 209–217, 1999.

[16] A. Lasota and M. C. Mackey, *Chaos, Fractals and Noise*, 1994.

[17] A. Mauroy and I. Mezić, "Global Stability Analysis Using the Eigenfunctions of the Koopman Operator," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3356–3369, 2016.

[18] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos*, vol. 22, no. 4, 2012.

[19] R. Mohr, "Spectral properties of the Koopman operator in the analysis of nonstationary dynamical systems," Ph.D. dissertation, 2014.

[20] Y. Lan and I. Mezić, "Linearization in the large of nonlinear systems and Koopman operator spectrum," *Physica D: Nonlinear Phenomena*, vol. 242, no. 1, pp. 42–53, 2013.

[21] J. D. Meiss, "5. Invariant Manifolds," *Differential Dynamical Systems*, pp. 165–195, 2007.

[22] B. Yi and I. R. Manchester, "On the Equivalence of Contraction and Koopman Approaches for Nonlinear Stability and Control," in *IEEE Conference on Decision and Control*, 2021, pp. 4609–4614.

[23] T. Teshima, I. Ishikawa, K. Tojo, K. Oono, M. Ikeda, and M. Sugiyama, "Coupling-based invertible neural networks are universal diffeomorphism approximators," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 3362–3373.

[24] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[25] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte, "Data-driven approximation of the Koopman generator: Model reduction, system identification, and control," *Physica D: Nonlinear Phenomena*, vol. 406, 2020.