Proceedings of Machine Learning Research vol 144:1–20, 2022

KoopmanizingFlows: Diffeomorphically Learning Stable Koopman Operators

Petar Bevanda[1] PETAR.BEVANDA@TUM.DE
Max Beier[1] MAX.BEIER@TUM.DE
Sebastian Kerz[2] SEBASTIAN.KERZ@TUM.DE
Armin Lederer[1] ARMIN.LEDERER@TUM.DE
Stefan Sosnowski[1] SOSNOWSKI@TUM.DE
Sandra Hirche[1] HIRCHE@TUM.DE
Department of Electrical and Computer Engineering
Technical University of Munich
D-80333 Munich, Germany
[1] *Chair of Information-oriented Control*
[2] *Chair of Automatic Control Engineering*

Abstract

We propose a novel framework for constructing linear time-invariant (LTI) models for data-driven representations of the Koopman operator for a class of stable nonlinear dynamics. The Koopman operator (generator) lifts a finite-dimensional nonlinear system to a possibly infinite-dimensional linear feature space. To utilize it for modeling, one needs to discover finite-dimensional representations of the Koopman operator. Learning suitable features is challenging, as one needs to learn LTI features that are both *Koopman-invariant* (evolve linearly under the dynamics) as well as *relevant* (spanning the original state) - a generally unsupervised learning task. For a theoretically well-founded solution to this problem, we propose learning Koopman-invariant coordinates by composing a diffeomorphic learner with a lifted aggregate system of a latent linear model. Using an unconstrained parameterization of stable matrices along with the aforementioned feature construction, we learn the Koopman operator features without assuming a predefined library of functions or knowing the spectrum, while ensuring stability regardless of the operator approximation accuracy. We demonstrate the superior efficacy of the proposed method in comparison to a state-of-the-art method on the well-known LASA handwriting dataset.

Keywords: Koopman operator, safe learning, learning of dynamical systems

1. Introduction

Global linearization methods for nonlinear systems inspired by the infinite-dimensional, *linear* Koopman operator (Koopman, 1931) have received increased attention for modeling nonlinear dynamics in recent years. By lifting a finite-dimensional nonlinear system to a higher-dimensional linear operator representation, superior complexity-accuracy balance compared to conventional nonlinear modeling is possible through the use of efficient linear techniques for prediction, analysis and control (Bevanda et al., 2021).

To strike a meaningful accuracy-complexity balance, finite-dimensional Koopman operator dynamical models are required to be long-term accurate with finite amount of features and data. Researchers aiming to achieve this property are faced with an unsupervised learning problem that involves learning linear time-invariant features that are both *Koopman-invariant*, i.e., their evolution remains in the span of the features, as well as *relevant* enough to (almost) fully span the original state - reconstructing it in a linear fashion.

As solving the aforementioned unsupervised learning task is challenging, the majority of works predicate on lifting the original nonlinear system to a projection of the operator onto predetermined features - akin to Galerkin methods - using well-known EDMD (Williams et al., 2015a) and its different variants (Williams et al., 2015b; Huang and Vaidya, 2018; Haseli and Cortes, 2021). However, presupposing a suitable basis of functions is a very strong assumption for linear time-invariant prediction - leading to only locally accurate models. Other approaches learn the features simultaneously (Li et al., 2017) or in a decoupled manner (Lian and Jones, 2019) leveraging the expressive power of neural networks or kernel methods, but often lack theoretical justification. Instead of arbitrary feature maps, Korda and Mezić (2020) learn the eigenfunctions of the operator for linear prediction - which is a promising proposition. Still, this approach leaves a gap between the operator-theoretic considerations and the data-driven realization.

Nevertheless, the sole expressivity of the learning methods does not immediately lead to reliability in constructing Koopman-invariant coordinates as the unsupervised problem of learning Koopman operator dynamical models requires certain structure to be well-posed. For dissipative systems, a way to improve the reliability of learning these models is enforcing stability. Nonetheless, most approaches do not impose such a constraint with guarantees. Notably, the recent work of Fan et al. (2021) demonstrates improved performance through stability guarantees in a fully data-driven manner. However, it's state reconstruction is done in a nonlinear fashion, limiting its practicability compared to immediate nonlinear modeling.

The contribution of this paper is the development of a novel *principled* framework for learning provably stable Koopman operator dynamical models, ensuring the system- and operator-theoretic considerations are fully embedded in our learning approach. Furthermore, it is fully data-driven as the features and the operator spectrum are learned simultaneously. We construct Koopman-invariant coordinates for a nonlinear system by composing a diffeomorphic learner with a lifted aggregate system of a latent linear model. Using an unconstrained parameterization of stable matrices, along with the aforementioned feature construction, we ensure stability regardless of the operator approximation accuracy. To the best of our knowledge, this is the first fully-data driven framework where the learning approach is well-founded in an operator- and system-theoretic sense. We demonstrate the superior performance of the proposed method in comparison to a state-of-the-art method on the well-known LASA handwriting dataset.

This paper is structured as follows. After the problem statement in Section 2, we present a novel data-driven framework - **K**oopmanizing**F**low **S**table **D**ynamical **S**ystems (KF-SDS) - for constructing stable, Koopman operator dynamical models in Section 3 which is followed by evaluation in Section 4 and a conclusion.

2. Problem Statement

Consider an unknown, continuous-time nonlinear dynamical system[1]

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

with continuous states on a compact set $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^d$ such that $\boldsymbol{f} \in C^2(\mathbb{X})$.

Assumption 1 *We assume the fixed point of (1) is globally exponentially stable.*

The above assumption is not restrictive in practice as it admits dynamical systems representing motion, e.g., human reaching movements (Khansari-Zadeh and Billard, 2011), or physical systems, e.g., a pendulum which rests in hanging position.

Due to their continuous-time nature, the dynamics are fully described by the forward-complete flow map (Bittracher et al., 2015) 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 (Angeli and Sontag, 1999). This flow map naturally induces the associated Koopman operator semigroup as defined in the following.

Definition 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 $\mathcal{K}^t_{\boldsymbol{f}} h = h \circ \boldsymbol{F}^t$.*

In simple terms, the operator applied to an observable function h at time t_0 moves it along the flow (2) as $\mathcal{K}^t_{\boldsymbol{f}} h(\boldsymbol{x}(t_0)) = h(\boldsymbol{x}(t_0 + t))$. Applied component-wise to the identity observable $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{x}$, it equals the flow (2). Crucially, every $\mathcal{K}^t_{\boldsymbol{f}}$ is a *linear*[2] operator. With a well-defined Koopman operator semigroup, we introduce its infinitesimal generator.

Definition 2 (Lasota and Mackey (1994)) *The evolution operator*

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

is the infinitesimal generator of the semigroup of Koopman operators $\{\mathcal{K}^t\}_{t\in\mathbb{R}_{+,0}}$.

The strength of the Koopman operator formalism is that it allows to decompose dynamics into linearly evolving coordinates, which naturally arise through the eigenfunctions of evolution operator $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$. These eigenfunctions are formally defined as follows.

Definition 3 *An observable $\phi \in C(\mathbb{X})$ is called an eigenfunction of $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$ if it satisfies $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}} \phi = \lambda \phi$, for an eigenvalue $\lambda \in \mathbb{C}$. The span of eigenfunctions ϕ of $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$ is denoted by $\boldsymbol{\Phi}$.*

With the above definitions, it is evident that the Koopman operator theory is inherently tied to the temporal evolution of dynamical systems (Bevanda et al., 2021). Moreover, due to Assumption 1, the Koopman operator generator has a pure point spectrum for the

[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/regularity order are denoted as L/C with the order (class) specified in their exponent. The Jacobian matrix of vector-valued map $\boldsymbol{\psi}$ evaluated at $\boldsymbol{x}$ is denoted as $\boldsymbol{J}_{\boldsymbol{\psi}}(\boldsymbol{x})$. The L^p-norm on a set $\mathbb{X}$ is denoted as $\|\cdot\|_{p,\mathbb{X}}$. Writing $\odot$ denotes the Hadamard product, *exp* pointwise exponential and $\circ$ function composition.

[2] Consider $h_1, h_2 \in C(\mathbb{X})$ and $\beta \in \mathbb{C}$. Then, using Definition 1, $\mathcal{K}_t(\beta h_1 + h_2) = (\beta h_1 + h_2) \circ \boldsymbol{F}_t = \beta h_1 \circ \boldsymbol{F}_t + h_2 \circ \boldsymbol{F}_t = \beta \mathcal{K}_t h_1 + \mathcal{K}_t h_2$.

dynamics (1) (Mauroy and Mezić, 2016). Therefore, for each observable h, there exists a sequence $v_j(h) \in \mathbb{C}$ of mode weights, such that we obtain the decomposition

$$\dot{h} = \mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}} h = \mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}} \left(\sum_{j=1}^{\infty} v_j(h) \phi_j \right) = \sum_{j=1}^{\infty} v_j(h) \lambda_j \phi_j. \tag{4}$$

Looking at the above decomposition, we can observe it is a superposition of infinitely many linear ODEs. With a slight abuse of operator notation, we can write the decomposition (4) compactly as $\dot{h} = \mathcal{V}_h \mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}} \boldsymbol{\Phi}$, where $\mathcal{V}_h$ is an operator projecting on the observable. As working with infinitely many states is not directly helpful in practice, we look to obtain a finite-dimensional representation. Since finding a meaningful finite-dimensional model is not analytically possible in general, we use data samples in order to obtain one.

Assumption 2 *A data-set of N input-output pairs $\mathbb{D}_N = \left\{ \boldsymbol{x}^{(i)}, \dot{\boldsymbol{x}}^{(i)} \right\}_{i=1}^{N}$ for the system (1) is available.*

Having measurements of the state and its time-derivative at disposal is a common assumption. If not directly available, the time-derivative of the state can be approximated through finite differences for practical applications. Based on the above data set, we consider the problem of learning a finite-dimensional model of (4) for the full-state observable $\boldsymbol{h}(\boldsymbol{x}){=}\boldsymbol{x}$, which can be posed as the optimization problem

$$\min_{\boldsymbol{A},\boldsymbol{C},\boldsymbol{\psi}(\cdot)} \sum_{i=1}^{N} \overbrace{\|\dot{\boldsymbol{x}}^{(i)} - \boldsymbol{C}\boldsymbol{A}\boldsymbol{\psi}(\boldsymbol{x}^{(i)})\|_2^2}^{\text{prediction}} + \overbrace{\|\boldsymbol{x}^{(i)} - \boldsymbol{C}\boldsymbol{\psi}(\boldsymbol{x}^{(i)})\|_2^2}^{\text{reconstruction}} \tag{5a}$$

$$\text{subject to:} \quad \boldsymbol{\psi} \in \boldsymbol{\Phi} \qquad \text{(Koopman-invariance)} \tag{5b}$$

$$\boldsymbol{A} \text{ is Hurwitz} \qquad \text{(stability)} \tag{5c}$$

with $\boldsymbol{\psi} = [\psi_1, \cdots, \psi_D]^\top$, $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ and $\boldsymbol{C} \in \mathbb{R}^{d \times D}$ providing a finite-dimensional representation in terms of a state-space model[3]

$$\boldsymbol{z}_0 = \boldsymbol{\psi}(\boldsymbol{x}(0)), \tag{6a}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z}, \tag{6b}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{C}\boldsymbol{z}. \tag{6c}$$

With this model, the nonlinearity of a d-dimensional ODE (1) is traded for a nonlinear “lift” (6a) of the initial condition $\boldsymbol{x}(0)$ to higher dimensional ($D \gg d$) Koopman-invariant coordinates (6b) such that the original state can be linearly reconstructed via (6c). Note that the Hurwitz condition (5c) generally suffices to ensure the asymptotic stability of the lifted dynamics. Moreover, (5) requires to learn an arbitrary amount of Koopman-invariant features directly instead of only finding ones that lie in a heuristically predetermined dictionary of functions. Thus, the sole error source in the resulting system (6) is due to the finite truncation of the infinite sum in (4). This is crucial for long-term accurate linear prediction, especially when, e.g., the model (6) is used as a motion generator (Khansari-Zadeh and Billard, 2012) for a system with safety-critical operation limits.

[3] Without loss of generality, generalized eigenfunctions, eigenvalues and modes are recovered with a similarity transform as the Koopman invariant-subspace of the generalized eigenfunctions of an arbitrary matrix is identical to that of its Jordan normal form (Korda and Mezić, 2020).

3. Diffeoorphicaly Learning Koopman-invariant Coordinates

3.1. Construction of Lifting Functions

For constructing a model satisfying (5b), we formally introduce a general description of Koopman-invariant coordinates. These coordinates do not necessarily have to be (generalized) eigenfunctions, but they need to be linear combinations thereof, as required by (5b). This property can be reformulated as a partial differential equation, as shown in the following lemma[4].

Lemma 4 *Consider the system* (1)*, a matrix* $\boldsymbol{A} \in \mathbb{R}^{D\times D}$ *and a finite set of features* $\boldsymbol{\psi} := [\psi_1(\boldsymbol{x}), \ldots, \psi_D(\boldsymbol{x})]^\top$ *with* $\psi_i(\boldsymbol{x}) \in C^1(\mathbb{X})$ *on a compact set* $\mathbb{X}$*. If this feature set solves the following linear partial differential equation (PDE)*

$$\boldsymbol{J}_{\boldsymbol{\psi}}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{\psi}(\boldsymbol{x}), \tag{7}$$

the features are admissible Koopman-invariant coordinates satisfying (5b)*.*

Instead of directly attempting to find solutions to the PDE of Lemma 4, we use a diffeomorphic relation to a latent linear model to obtain solutions for (7), providing us with Koopman-invariant lifting coordinates that fulfill (5b). The overview of our construction can be found in Figure 1.

Definition 5 *Vector fields* $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ *and* $\dot{\boldsymbol{y}} = \boldsymbol{t}(\boldsymbol{y})$ *are diffeomorphic, or smoothly equivalent, if there exists a diffeomorphism* $\boldsymbol{d} : \mathbb{R}^d \mapsto \mathbb{R}^d$ *such that* $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{J}_{\boldsymbol{d}}^{-1}(\boldsymbol{x})\boldsymbol{t}(\boldsymbol{d}(\boldsymbol{x}))$*.*

In essence, diffeomorphic systems have equivalent dynamics just in different coordinates. For example, the eigenvalues of corresponding equilibria are the same (Kuznetsov, 2004). This is what we exploit as the system (1) is diffeomorphic to a latent linear system $\dot{\boldsymbol{y}} = \tilde{\boldsymbol{A}}\boldsymbol{y}$ under Assumption 1 (Lan and Mezić, 2013). Standalone, the diffeomorphic relation directly provides a nonlinear model

$$\boldsymbol{y}_0 = \boldsymbol{d}(\boldsymbol{x}(0)), \tag{8a}$$

$$\dot{\boldsymbol{y}} = \tilde{\boldsymbol{A}}\boldsymbol{y}, \tag{8b}$$

$$\boldsymbol{x} = \boldsymbol{d}^{-1}(\boldsymbol{y}), \tag{8c}$$

for (1). It is straightforward to see that the diffeomorphism $\boldsymbol{d}$ satisfies the conditions of Lemma 5, such that $\boldsymbol{\psi} = \boldsymbol{d}$ are features satisfying condition (5b). Nevertheless, (8) is still a nonlinear model after the initial transformation (8a), while the goal is to find a linear reconstruction map (6c) for a finite representation of (4). For achieving this linear reconstruction, we need to allow the latent dynamical system to have a dimension $D \gg d$, which cannot be achieved directly with diffeomorphisms since they preserve dimensionality. Therefore, we propose to "lift" the diffeomorphic features $\boldsymbol{d}$ to a higher dimensional space to exploit their natural satisfaction of (5b). This requires the lifting to maintain Koopman-invariance, which we achieve through a monomial expansion inspired by the notion of linear tensor systems (Zeng, 2018). Here, the idea is to define monomial coordinates based on the latent vector $\boldsymbol{d}(\boldsymbol{x}) = \boldsymbol{y} = [y_1, \ldots, y_d]^\top$ through $y^{\boldsymbol{\alpha}} = y_1^{\alpha_1} y_2^{\alpha_2} \cdots y_d^{\alpha_d}$, where $\boldsymbol{\alpha} \in \mathbb{N}_0^d$ is a multi-index. Then, we obtain a lifted coordinate vector by concatenating all monomials $y^{\boldsymbol{\alpha}}$ up to order $\|\boldsymbol{\alpha}\|_1 = \alpha_1 + \cdots + \alpha_d \leq \overline{p}$ in a lexicographical ordering in a vector $\boldsymbol{y}^{[\overline{p}]}$. Due

[4] Proofs for all theoretical results can be found in Appendix A

$$\begin{tikzcd}
\mathbb{X} \arrow[r, "\boldsymbol{d}"] \arrow[d, "\boldsymbol{f}"', "(1)"] \arrow[rr, bend left, "\boldsymbol{\psi}", "\text{Prop. } 7"'] & \mathbb{Y} \arrow[r, "\boldsymbol{\varrho}\colon\, \boldsymbol{y}\mapsto\boldsymbol{y}^{[\overline{p}]}", "\text{Lem. } 6"'] \arrow[d, "\tilde{\boldsymbol{A}}"'] & \mathbb{Z} \arrow[d, "\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})"'] \\
\mathcal{T}_{\boldsymbol{x}}\mathbb{X} \arrow[r, "\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}}"] & \mathcal{T}_{\boldsymbol{y}}\mathbb{Y} & \mathcal{T}_{\boldsymbol{z}}\mathbb{Z} \arrow[ll, bend left, dashed, "\boldsymbol{C}"]
\end{tikzcd}$$

Figure 1: The diagram of our construction for learning model of the form (6a)-(6c) with the construction pathway in bold and the maps to be learned in magenta. The sets $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ correspond to the immediate state-space, latent and lifted linear model space, respectively; with corresponding tangent spaces denotes as $\mathcal{T}_{\boldsymbol{x}}\mathbb{X}, \mathcal{T}_{\boldsymbol{y}}\mathbb{Y}, \mathcal{T}_{\boldsymbol{z}}\mathbb{Z}$.

to the construction of this vector, it inherits the linear dynamical system description from $\boldsymbol{y} = \boldsymbol{d}(\boldsymbol{x})$ in (8b), as shown in the following lemma.

Lemma 6 *There exists a matrix $\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}}) \in \mathbb{R}^{D\times D}$, $D = ((d+\overline{p})!/(d!\overline{p}!)) - 1$, such that the evolution of the lifted coordinates $\boldsymbol{y}^{[\overline{p}]}$ is described by the linear ordinary differential equation*

$$\frac{d}{dt}\boldsymbol{y}^{[\overline{p}]} = \boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})\boldsymbol{y}^{[\overline{p}]}. \tag{9}$$

The matrix $\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})$ in (9) can be straightforwardly calculated given $\tilde{\boldsymbol{A}}$, such that it provides a low-dimensional parameterization of the latent linear dynamics. Moreover, this monomial lifting of the latent linear syetem preserves the Koopman-invariance of the diffeomorphism $\boldsymbol{d}$ and satisfies (5b) as shown in the following proposition.

Proposition 7 *Assume the linear system $\dot{\boldsymbol{y}} = \tilde{\boldsymbol{A}}\boldsymbol{y}$ is smoothly equivalent to system (1) via a diffeomorphism $\boldsymbol{d}$ such that $\boldsymbol{y} = \boldsymbol{d}(\boldsymbol{x})$. Then the lifted features $\boldsymbol{\psi} = \boldsymbol{d}^{[\overline{p}]}$ satisfy (5b), i.e., $\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}) = \boldsymbol{y}^{[\overline{p}]}$ are Koopman-invariant coordinates and define a latent linear system*

$$\boldsymbol{z}_0 = \boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}(0)), \tag{10a}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})\boldsymbol{z}. \tag{10b}$$

Due the above result, learning Koopman-invariant features reduces to learning a diffeomorphism, which allows us to replace the constraint (5b) by the condition

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{J}_{\boldsymbol{d}}^{-1}(\boldsymbol{x})\tilde{\boldsymbol{A}}\boldsymbol{d}(\boldsymbol{x}), \tag{11}$$

which ensures the smooth equivalence between $\boldsymbol{f}$ and $\boldsymbol{A}$. This can be interpreted as turning an unsupervised to a supervised learning problem, since we generally do not have access to Koopman invariant coordinates (5b), while it is straightforward to check (11) given a diffeomorphism $\boldsymbol{d}$. Therefore, Proposition 7 allows us to simplify problem (5) to

$$\min_{\tilde{\boldsymbol{A}},\boldsymbol{C},\boldsymbol{d}(\cdot)} \sum_{i=1}^{N} \|\dot{\boldsymbol{x}}^{(i)} - \boldsymbol{C}\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})\boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}^{(i)})\|_2^2 + \|\boldsymbol{x}^{(i)} - \boldsymbol{C}\boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}^{(i)})\|_2^2 \tag{12a}$$

$$\text{subject to:} \quad \boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}}) \text{ is Hurwitz} \qquad \text{(stability)} \tag{12b}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{J}_{\boldsymbol{d}}^{-1}(\boldsymbol{x})\tilde{\boldsymbol{A}}\boldsymbol{d}(\boldsymbol{x}). \qquad \text{(smooth equivalence)} \tag{12c}$$

3.2. Parameterizing Stable System Matrices

To simplify the computations involved for satisfying (12b), we propose to employ an unconstrained parameterization of stable matrices akin to Fan et al. (2021). As the latent dynamics is parameterized in terms of low-dimensional matrices $\tilde{\boldsymbol{A}}$, we utilize an unconstrained parameterization of all Hurwitz $\tilde{\boldsymbol{A}}$ matrices, described by the following lemma.

Lemma 8 *Consider real-valued matrices $\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R} \in \mathbb{R}^{n\times n}$, a positive constant $\epsilon \in \mathbb{R}_+$, and the matrix $\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R}) \in \mathbb{R}^{n\times n}$ defined as*

$$\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R}) = \left(\boldsymbol{N}\boldsymbol{N}^\top + \epsilon\boldsymbol{I}\right)^{-1}\left(-\boldsymbol{Q}\boldsymbol{Q}^\top - \epsilon\boldsymbol{I} + \frac{1}{2}\left(\boldsymbol{R} - \boldsymbol{R}^\top\right)\right). \tag{13}$$

($\Rightarrow$) For all $\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R},\epsilon$ as above, $\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R})$ is Hurwitz.

($\Leftarrow$) For all Hurwitz matrices $\boldsymbol{\mathcal{A}}$, $\exists$ $\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R},\epsilon$ as above, such that $\boldsymbol{\mathcal{A}} = \tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R})$.

With $\epsilon > 0$ fixed and sufficiently small, $\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R}$ from Lemma 8 serve as an unconstrained parameterization of (practically) all Hurwitz matrices. For $\epsilon \to 0^+$, the space of all Hurwitz matrices is covered. Thus, we can optimize over the low-dimensional matrices $\boldsymbol{N}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ without worrying about the stability condition (12b) as it is guaranteed by construction. Due to Lemma 6, this stability extends to the lifted system matrix $\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R}))$, such that we can reformulate the optimization problem (5) as shown in the following theorem.

Theorem 9 *The minimizers $\hat{\boldsymbol{N}},\hat{\boldsymbol{Q}},\hat{\boldsymbol{R}},\hat{\boldsymbol{C}},\hat{\boldsymbol{d}}(\cdot)$ of the optimization problem*

$$\min_{\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R},\boldsymbol{C},\boldsymbol{d}(\cdot)} \sum_{i=1}^{N} \|\dot{\boldsymbol{x}}^{(i)} - \boldsymbol{C}\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R}))\boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}^{(i)})\|_2^2 + \|\boldsymbol{x}^{(i)} - \boldsymbol{C}\boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}^{(i)})\|_2^2 \tag{14a}$$

$$\text{subject to}: \quad \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{J}_{\boldsymbol{d}}^{-1}(\boldsymbol{x})\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R})\boldsymbol{d}(\boldsymbol{x}) \qquad \textit{(smooth equivalence)} \tag{14b}$$

define a solution $\boldsymbol{\psi} = \hat{\boldsymbol{d}}^{[\overline{p}]}$, $\boldsymbol{A} = \boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}}(\hat{\boldsymbol{N}},\hat{\boldsymbol{Q}},\hat{\boldsymbol{R}}))$, $\boldsymbol{C} = \hat{\boldsymbol{C}}$ for the optimization problem (5) and thereby define a model of the form (6).

In principle, the above optimization problem admits a direct implementation using Lagrange multipliers (Nocedal and Wright, 2006). This, in turn, allows us to reformulate problem (14) in an unconstrained manner - simplifying the optimization task in principle.

3.3. Structured Relaxation of Exact Smooth Equivalence

To ease the use of standard training algorithms for expressive function approximators such as neural networks, we relax the optimization problem (14) by considering (14b) as an additional summand in the cost (14a), which can be considered as fixing the value of a

Lagrange multiplier to a specific value. This results in the unconstrained optimization problem

$$\min_{\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R},\boldsymbol{C},\boldsymbol{d}(\cdot)} \sum_{i=1}^{N} \|\dot{\boldsymbol{x}}^{(i)} - \boldsymbol{C}\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R}))\boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}^{(i)})\|_2^2 + \|\boldsymbol{x}^{(i)} - \boldsymbol{C}\boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x}^{(i)})\|_2^2 + \mathcal{L}_{\mathrm{SE}}(\boldsymbol{x}^{(i)},\dot{\boldsymbol{x}}^{(i)}), \tag{15}$$

where the cost

$$\mathcal{L}_{\mathrm{SE}}(\boldsymbol{x}^{(i)},\dot{\boldsymbol{x}}^{(i)}) = \|\dot{\boldsymbol{x}} - \boldsymbol{J}_{\boldsymbol{d}}^{-1}(\boldsymbol{x})\tilde{\boldsymbol{A}}(\boldsymbol{N},\boldsymbol{Q},\boldsymbol{R})\boldsymbol{d}(\boldsymbol{x})\|_2^2 + \|\boldsymbol{J}_{\boldsymbol{d}}(\boldsymbol{0}) - \boldsymbol{I}\|_2^2 + \|\boldsymbol{d}(\boldsymbol{0}) - \boldsymbol{0}\|_2^2 \tag{16}$$

replaces the constraint (14b). Note that the two last summands of $\mathcal{L}_{\mathrm{SE}}$ are not necessary in principle, but can be used to enforce near-identity of the diffeomorphism, thereby improving convergence in a local neighborhood of the equilibrium (Lan and Mezić, 2013, Theorem 2.3).

In order to finally solve (15), one needs to ensure the function approximator used for learning $\boldsymbol{d}$ is guaranteed to be a diffeomorphism. For this, we utilize coupling flow invertible neural networks (CF-INN). Although predominantly used as distribution estimators (Dinh et al., 2017), they can also be employed for regression, e.g., modeling stable nonlinear dynamics in immediate state-spaces (Rana et al., 2020). Moreover, CF-INN have been shown to exhibit universal approximation properties (Teshima et al., 2020), which allows to approximate a large class of diffeomorphisms arbitrarily well with respect to the L^p-/sup-norm.

For realizing complex diffeomorphisms, CF-INN successively compose simpler diffeomorphisms called *coupling layers* $\hat{\boldsymbol{d}}_{\boldsymbol{i}}$ using the fact that diffeomorphic maps are closed under composition, so that $\boldsymbol{y} = \hat{\boldsymbol{d}}(\boldsymbol{x}) = \hat{\boldsymbol{d}}_{\boldsymbol{k}} \circ \ldots \circ \hat{\boldsymbol{d}}_{\boldsymbol{1}}(\boldsymbol{x})$. Each coupling layer $\hat{\boldsymbol{d}}_{\boldsymbol{i}}$ is defined to couple a disjoint partition of the input $\boldsymbol{x} = [\boldsymbol{x}_a^\top, \boldsymbol{x}_b^\top]^\top$ with two subspaces $\boldsymbol{x}_{\boldsymbol{a}} \in \mathbb{R}^{d-n}$, $\boldsymbol{x}_{\boldsymbol{b}} \in \mathbb{R}^n$ where $n \in \mathbb{N}$ and $d \geq 2$, in a manner that ensures bijectivity. This can be realized via affine coupling flows (ACF), which have coupling layers

$$\hat{\boldsymbol{d}}_i(\boldsymbol{x}^{(i)}) = \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{a}}^{(i)} \\ \boldsymbol{x}_{\boldsymbol{b}}^{(i)} \odot exp(\boldsymbol{s}_{\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{a}}^{(i)})) + \boldsymbol{t}_{\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{a}}^{(i)}) \end{bmatrix} \tag{17}$$

with scaling functions $\boldsymbol{s}_i : \mathbb{R}^n \mapsto \mathbb{R}^{N-n}$ and translation functions $\boldsymbol{t}_i : \mathbb{R}^n \mapsto \mathbb{R}^{N-n}$ that can be chosen freely. The parameters of the diffeomorphic learner consist of the the weights and biases in the neural networks of the scaling and translation functions concatenated in parameters $\boldsymbol{w} = [\boldsymbol{w}_{\boldsymbol{s}_1}^\top, \boldsymbol{w}_{\boldsymbol{t}_1}^\top, \cdots, \boldsymbol{w}_{\boldsymbol{s}_k}^\top, \boldsymbol{w}_{\boldsymbol{t}_k}^\top]^\top$.

Since the ACF are constructed to be diffeomorphisms, we can optimize over the parameters $\boldsymbol{w}$ instead of diffeomorphisms in (12). Moreover, it allows us to guarantee the stability of systems (6) induced by the solutions of (12), as shown in the following theorem.

Theorem 10 *Consider diffeomorphisms* $\boldsymbol{d} = \hat{\boldsymbol{d}}_{\boldsymbol{k}} \circ \ldots \circ \hat{\boldsymbol{d}}_{\boldsymbol{1}}(\boldsymbol{x})$ *parameterized through coupling layers* (17)*, which are defined using continuously differentiable functions* $\boldsymbol{s}_i$*,* $\boldsymbol{t}_i$*.Then, every optimization problem* (15) *has a solution and yields a stable system* (6)*.*

This theorem allows to efficiently obtain approximate solutions to the optimization problem (5) in practice, since the differentiability condition for $\boldsymbol{s}_i$, $\boldsymbol{t}_i$ can be easily satisfied using neural networks with smooth activation functions. Therefore, it transforms the practically intractable problem (5) into an easily implementable deep learning problem. While we cannot ensure that the results of this deep learning problem are an exact solution to (5), Theorem 10 guarantees that these solutions yield stable systems (6).

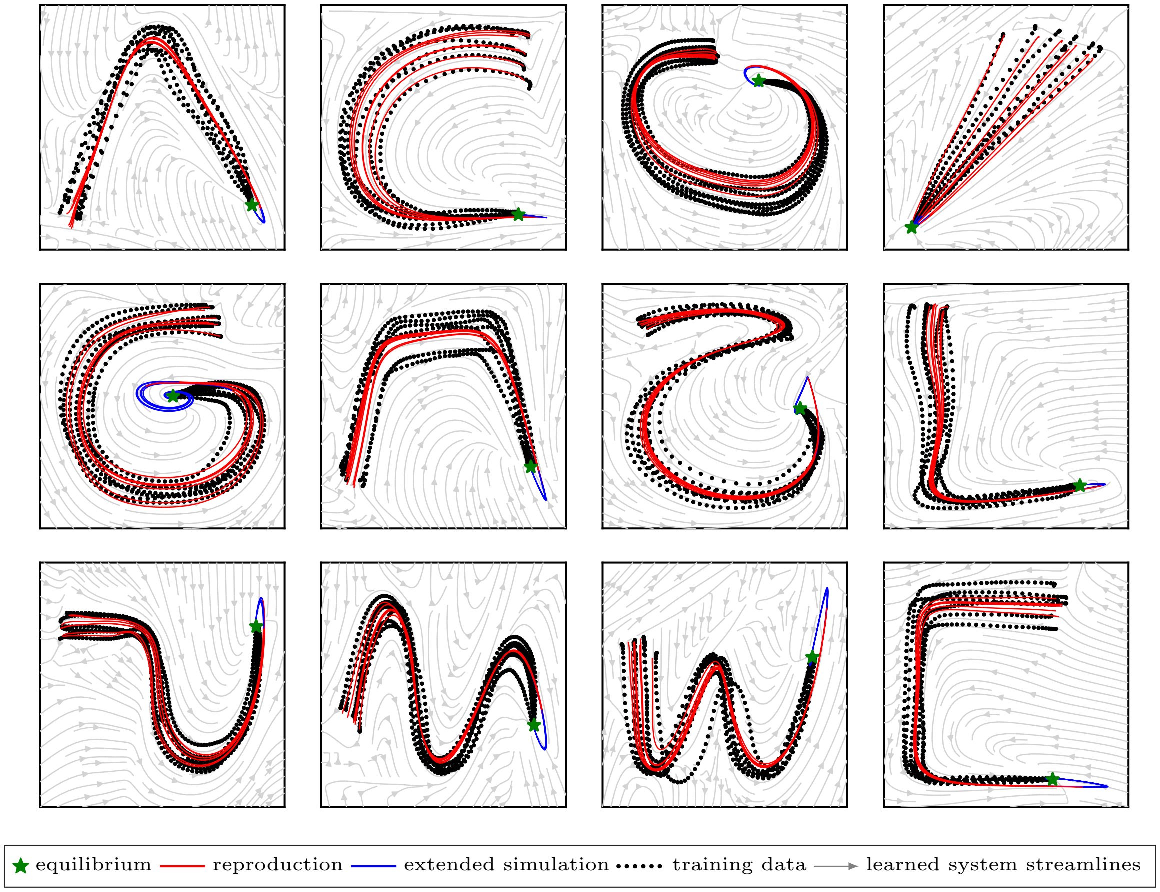

Figure 2: KF-SDS yields trajectories similar in shape to the real ones, demonstrating the lifting construction captures the geometry of the original state-space.

4. Evaluation

We evaluate our methods on the real-world LASA[5] handwriting dataset (Khansari-Zadeh and Billard, 2011). All data is scaled to the range $[-1, 1]^d$ before training. We have sampled 900 data points for each of the 7 demonstration trajectories of the dataset. The inputs and targets are the 2D position and velocity, respectively. For all experiments, ACF with 10 coupling layers are used to learn the diffeomorphisms. The neural networks for the scaling and translation functions in each of the affine coupling layers have 3 hidden layers, with 120 neurons, each with a smooth Exponential Linear Unit (ELU) as the activation function. The dimension of the lifting coordinates is $D = 44$ ($\bar{p} = 8$). Full batch learning is performed, employing the ADAM optimizer (Kingma and Ba, 2015). The reproduction of the demonstrated trajectories displayed in Figure 2 is simulated until five times the demonstration time, with a different coloring from the point when the demonstration time is exceeded. Due to the deterministic nature of KF-SDS, cross-sections of the demonstrations

[5] https://cs.stanford.edu/people/khansari/download.html

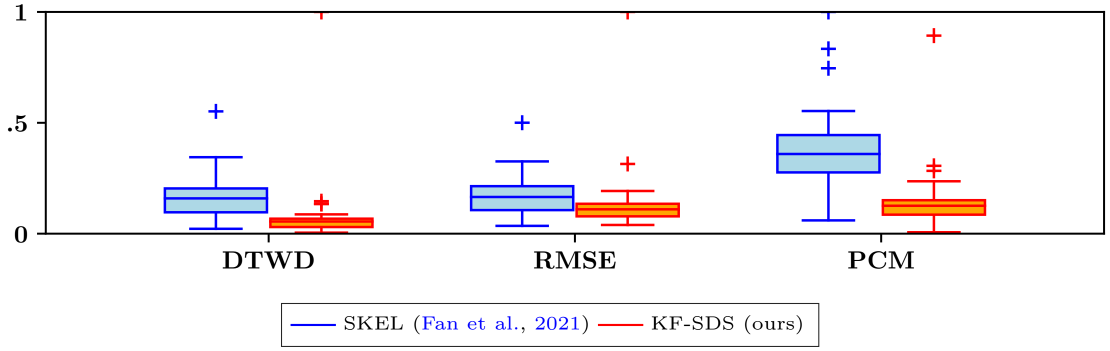

Figure 3: KF-SDS shows superior performance, especially in capturing the accuracy of the shapes. Each metric is normalized to lie in the range $[0, 1]$ for ease of comparison.

lead to a contraction to a mean trajectory when reproduced. In order to evaluate our performance with respect to the related work of Fan et al. (2021), we compare our position prediction in terms of dynamic time warping distance (DTWD) (Berndt and Clifford, 1994), root mean squared error (RMSE) and partial curve matching (PCM) (Jekel et al., 2019). The statistics for each of the frameworks in the respective metrics are visualized in Figure 3. In comparison to SKEL[6], KF-SDS shows superior performance in all metrics - especially ones strongly related to the accuracy of the state-space geometry such as DTWD and PCM. This is to be expected as KF-SDS is geared towards directly learning features that lie in span of (generalized) eigenfunctions and inherently describe the state-space geometry (Mezić, 2019). Furthermore, the performance of SKEL deteriorates in the long-term due to a nonlinear reconstruction map, which can cause the equilibrium point of the model to not lie at the end-point of the demonstrated movement[7]. Therefore, the proposed KF-SDS framework shows how a theoretically well-founded construction of lifting features results in significant performance advantages.

5. Conclusion

We have presented KF-SDS, an approach for fully data-driven learning of stable Koopman operator models with linear prediction and reconstruction. Our results demonstrate improved performance compared to related work with nonlinear original state reconstruction even though employing the more practicable linear reconstruction. An experimental evaluation on a handwriting dataset shows the superior efficacy of our structured learning approach.

Acknowledgments

We thank Samuel Tesfazgi for useful comments in preparing this manuscript and Fletcher Fan for kindly providing us with the source code for SKEL (Fan et al., 2021). This work was supported by the 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).

[6] https://github.com/fletchf/skel.git, Last accessed: Dec. 2, 2021

[7] Additional details on the evaluation part can be found in Appendix B.

References

David Angeli and Eduardo D. Sontag. Forward completeness, unboundedness observability, and their Lyapunov characterizations. *Systems and Control Letters*, 38(4-5):209–217, 1999. ISSN 01676911. doi: 10.1016/S0005-1098(99)00037-0.

Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, page 359370. AAAI Press, 1994.

Petar Bevanda, Stefan Sosnowski, and Sandra Hirche. Koopman operator dynamical models: Learning, analysis and control. *Annual Reviews in Control*, 52:197–212, 2021. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2021.09.002. URL https://www.sciencedirect.com/science/article/pii/S1367578821000729.

Andreas Bittracher, Péter Koltai, and Oliver Junge. Pseudogenerators of spatial transfer operators. *SIAM Journal on Applied Dynamical Systems*, 14(3):1478–1517, 2015. doi: 10.1137/14099872X.

Erik M. Bollt, Qianxiao Li, Felix Dietrich, and Ioannis Kevrekidis. On matching, and even rectifying, dynamical systems through Koopman operator eigenfunctions. *SIAM Journal on Applied Dynamical Systems*, 17(2):1925–1960, 2017. doi: 10.1137/17M116207X.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

Fletcher Fan, Bowen Yi, David Rye, Guodong Shi, and Ian R. Manchester. Learning Stable Koopman Embeddings. 2021. URL http://arxiv.org/abs/2110.06509.

Masih Haseli and Jorge Cortes. Learning Koopman Eigenfunctions and Invariant Subspaces from Data: Symmetric Subspace Decomposition. *IEEE Transactions on Automatic Control*, 9286(c):1–1, 2021. ISSN 0018-9286. doi: 10.1109/TAC.2021.3105318. URL http://arxiv.org/abs/1909.01419https://ieeexplore.ieee.org/document/9516947/.

Bowen Huang and Umesh Vaidya. Data-driven approximation of transfer operators: Naturally structured dynamic mode decomposition. In *2018 Annual American Control Conference (ACC)*, pages 5659–5664, 2018. doi: 10.23919/ACC.2018.8431409.

Charles F. Jekel, Gerhard Venter, Martin P. Venter, Nielen Stander, and Raphael T. Haftka. Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. *International Journal of Material Forming*, 12(3):355–378, May 2019. ISSN 1960-6214. doi: 10.1007/s12289-018-1421-8. URL https://doi.org/10.1007/s12289-018-1421-8.

Charles H. Jones. Generalized hockey stick identities and N-dimensional blockwalking. *Fibonacci Quarterly*, 34(3):280–288, 1996. ISSN 00150517.

S. Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011. doi: 10.1109/TRO.2011.2159412.

Seyed Mohammad Khansari-Zadeh and Aude Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454, May 2012. ISSN 1573-7527. doi: 10.1007/s10514-012-9287-y. URL https://doi.org/10.1007/s10514-012-9287-y.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

B. O. Koopman. Hamiltonian Systems and Transformation in Hilbert Space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315–318, 1931.

Milan Korda and Igor Mezić. Optimal construction of koopman eigenfunctions for prediction and control. *IEEE Transactions on Automatic Control*, 65(12):5114–5129, 2020. doi: 10.1109/TAC.2020.2978039.

Yuri A. Kuznetsov. *Topological Equivalence, Bifurcations, and Structural Stability of Dynamical Systems*, pages 39–76. Springer New York, New York, NY, 2004. ISBN 978-1-4757-3978-7. doi: 10.1007/978-1-4757-3978-7_2. URL https://doi.org/10.1007/978-1-4757-3978-7_2.

Yueheng Lan and Igor Mezić. Linearization in the large of nonlinear systems and Koopman operator spectrum. *Physica D: Nonlinear Phenomena*, 242(1):42–53, 2013. doi: 10.1016/j.physd.2012.08.017.

Andrzej Lasota and Michael C. Mackey. *Chaos, Fractals and Noise*. 1994. doi: 10.1198/tech.2006.s351.

Qianxiao Li, Felix Dietrich, Erik M. Bollt, and Ioannis G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos*, 27(10), 2017. doi: 10.1063/1.4993854.

Yingzhao Lian and Colin N. Jones. Learning Feature Maps of the Koopman Operator: A Subspace Viewpoint. In *2019 IEEE 58th Conference on Decision and Control*, pages 860–866, 2019. ISBN 978-1-7281-1398-2. doi: 10.1109/CDC40024.2019.9029189.

Alexandre Mauroy and Igor Mezić. Global Stability Analysis Using the Eigenfunctions of the Koopman Operator. *IEEE Transactions on Automatic Control*, 61(11):3356–3369, 2016. doi: 10.1109/TAC.2016.2518918.

Igor Mezić. Spectrum of the Koopman Operator, Spectral Expansions in Functional Spaces, and State-Space Geometry. *Journal of Nonlinear Science volume*, pages 1–40, 2019. doi: 10.1007/s00332-019-09598-5.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

Muhammad Asif Rana, Anqi Li, Dieter Fox, Byron Boots, Fabio Ramos, and Nathan Ratliff. Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 630–639. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/rana20a.html.

Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3362–3373. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/2290a7385ed77cc5592dc2153229f082-Paper.pdf.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science volume*, 25(6):1307–1346, 2015a. doi: 10.1007/s00332-015-9258-5.

Matthew O. Williams, Clarence W. Rowley, and Ioannis G. Kevrekidis. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015b. doi: 10.3934/jcd.2015005.

Bowen Yi and Ian R. Manchester. On the equivalence of contraction and Koopman approaches for nonlinear stability and control, 2021.

Shen Zeng. On systems theoretic aspects of koopman operator theoretic frameworks. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6422–6427, 2018. doi: 10.1109/CDC.2018.8619090.

Appendix A. Proofs of Theoretical Results

A.1. Lifting Functions

Proof of Lemma 4 By construction, we seek a vector $\boldsymbol{\psi}$ of observable functions that is governed by a linear differential equation, meaning it satisfies

$$\frac{d}{dt}\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{\psi}(\boldsymbol{x}). \tag{18}$$

By Definition 2, the infinitesimal-time evolution operator, Koopman generator, acts on the observables vectors such that

$$\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}\boldsymbol{\psi}(\boldsymbol{x}) = \lim_{t\to 0^+} \frac{\boldsymbol{\psi}\left(\boldsymbol{F}^t\left(\boldsymbol{x}_0\right)\right) - \boldsymbol{\psi}\left(\boldsymbol{x}_0\right)}{t} = \lim_{t\to 0^+} \frac{\boldsymbol{\psi}(\boldsymbol{x}(t)) - \boldsymbol{\psi}\left(\boldsymbol{x}_0\right)}{t}, \tag{19}$$

directly following the definition of the infinitesimal generator (Lasota and Mackey, 1994). As $\boldsymbol{\psi}$ is a vector of continuously differentiable functions on a compact set $\mathbb{X}$, such that

$\psi_i \in C^1(\mathbb{X})$[8], applying the mean value theorem to (19) gives

$$\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}} \boldsymbol{\psi}(\boldsymbol{x}) = \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{20}$$

$$= \boldsymbol{J}_{\boldsymbol{\psi}}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}), \tag{21}$$

equivalent to the Lie derivative. Equation (18) together with (20) give the following PDE

$$\boldsymbol{J}_{\boldsymbol{\psi}}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) = \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{22}$$

$$= \boldsymbol{A}\boldsymbol{\psi}(\boldsymbol{x}) \tag{23}$$

proving the lemma. ■

Proof of Lemma 6 By examining the dynamics of a monomial corresponding to the multi-index $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_d]^\top$ with order $\|\boldsymbol{\alpha}\|_1 = p$ (Zeng, 2018), we see that

$$\frac{d}{dt} y^{\boldsymbol{\alpha}} = \sum_{i=1}^{d} \alpha_i y_i^{\alpha_i - 1} \left(\prod_{j \neq i} y_j^{\alpha_j} \right) \dot{y}_i \tag{24}$$

holds. Now, let us denote $\boldsymbol{y}^{[p]}$ as vector of all $y^{\boldsymbol{\alpha}}$ of all different multi-indices of order $\|\boldsymbol{\alpha}\|_1 = p$. For example, by considering a monomial mapping $\boldsymbol{\varrho} : \boldsymbol{y} \mapsto \boldsymbol{y}^{[2]}$ with multi-indices $\boldsymbol{\alpha}_1 = [2, 0]^\top$, $\boldsymbol{\alpha}_2 = [1, 1]^\top$, $\boldsymbol{\alpha}_3 = [0, 2]^\top$ the following

$$\frac{d}{dt}\boldsymbol{y}^{[2]} = \begin{bmatrix} \frac{d}{dt} y^{\boldsymbol{\alpha}_1} \\ \frac{d}{dt} y^{\boldsymbol{\alpha}_2} \\ \frac{d}{dt} y^{\boldsymbol{\alpha}_3} \end{bmatrix} = \frac{\partial \boldsymbol{\varrho}}{\partial \boldsymbol{y}}(\boldsymbol{y})\dot{\boldsymbol{y}} \tag{25}$$

$$= \begin{bmatrix} 2y_1 & 0 \\ y_1 & y_2 \\ 0 & 2y_2 \end{bmatrix} \tilde{\boldsymbol{A}}\boldsymbol{y} \tag{26}$$

$$= \begin{bmatrix} 2\tilde{a}_{11} & 2\tilde{a}_{12} & 0 \\ \tilde{a}_{21} & \tilde{a}_{11} + \tilde{a}_{22} & \tilde{a}_{12} \\ 0 & 2\tilde{a}_{21} & 2\tilde{a}_{22} \end{bmatrix} \begin{bmatrix} y^{[2,0]^\top} \\ y^{[1,1]^\top} \\ y^{[0,2]^\top} \end{bmatrix} \tag{27}$$

$$= \begin{bmatrix} 2\tilde{a}_{11} & 2\tilde{a}_{12} & 0 \\ \tilde{a}_{21} & \tilde{a}_{11} + \tilde{a}_{22} & \tilde{a}_{12} \\ 0 & 2\tilde{a}_{21} & 2\tilde{a}_{22} \end{bmatrix} \begin{bmatrix} y_1^2 \\ y_1 y_2 \\ y_2^2 \end{bmatrix} \tag{28}$$

$$= \boldsymbol{A}_{[2]}(\tilde{\boldsymbol{A}})\boldsymbol{y}^{[2]} \tag{29}$$

$$\tag{30}$$

demonstrates that $\boldsymbol{y}^{[2]}$ has dynamics linearly dependent on $\tilde{\boldsymbol{A}}$. This follows in a straight forward manner for the dynamics of any other $\boldsymbol{y}^{[p]}$ (Zeng, 2018). Thus, we can write their dynamics as a set of linear ordinary differential equations

$$\frac{d}{dt}\boldsymbol{y}^{[p]} = \boldsymbol{A}_{[p]}(\tilde{\boldsymbol{A}})\boldsymbol{y}^{[p]}. \tag{31}$$

[8] For a more general $\mathbb{X}$, the result holds for $h_i(\boldsymbol{x}) \in C^2(\mathbb{X})$ Bollt et al. (2017)

Since all $\boldsymbol{y}^{[p]}$ systems $p \in \mathbb{N}$ are decoupled from each other, their concatenation up to order $\overline{p}$ as

$$\frac{d}{dt}\boldsymbol{y}^{[\overline{p}]} = \boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})\boldsymbol{y}^{[\overline{p}]} \tag{32}$$

with $\boldsymbol{y}^{[\overline{p}]} = [\boldsymbol{y}^{[1]^\top}, \dots, \boldsymbol{y}^{[p]^\top}, \dots, \boldsymbol{y}^{[\overline{p}]^\top}]^\top$ and $\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}}) = \mathrm{diag}\{\boldsymbol{A}_{[1]}(\tilde{\boldsymbol{A}}), \dots, \boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})\} \in \mathbb{R}^{D\times D}$ while still remaining a set of linear ordinary differential equations. Given that every $\boldsymbol{y}^{[p]}$ has as many elements as there are combinations with replacement of d elements and p samples, the total amount of concatenated coordinates up to order $\overline{p}$ equals to

$$D = \sum_{p=1}^{\overline{p}} \left(\begin{array}{c} p+d-1 \\ p \end{array} \right) = \left(\begin{array}{c} d+\overline{p} \\ d \end{array} \right) - 1. \tag{33}$$

The above sum is simplified using the "hockey-stick" identity (Jones, 1996). This proves the concatenation up to order $\overline{p}$ leads to an extended system $\dot{\boldsymbol{y}}^{[\overline{p}]} = \boldsymbol{A}_{[p]}(\tilde{\boldsymbol{A}})\boldsymbol{y}^{[\overline{p}]}$ spanning invariant subspaces of $\dot{\boldsymbol{y}} = \tilde{\boldsymbol{A}}\boldsymbol{y}$ of size D. ■

Proof of Proposition 7 Consider vector fields $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ and $\dot{\boldsymbol{y}} = \tilde{\boldsymbol{A}}\boldsymbol{y}$ smoothly equivalent through a diffeomorphism $\boldsymbol{y} = \boldsymbol{d}(\boldsymbol{x})$. A simple chain of equalities

$$\begin{aligned} \mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}\boldsymbol{d}(\boldsymbol{x}) &\overset{Lem.4}{=} \boldsymbol{J}_{\boldsymbol{d}}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) & (34)\\ &\overset{Def.5}{=} \tilde{\boldsymbol{A}}\boldsymbol{d}(\boldsymbol{x}) & (35)\\ &= \mathcal{G}_{\mathcal{K}_{\tilde{\boldsymbol{A}}}}\boldsymbol{y} & (36) \end{aligned}$$

shows Koopman-invariant subspaces related though a diffeomorphism of the respective vector fields' infinitesimal generators evolve linearly with the same $\tilde{\boldsymbol{A}}$. Following the construction of Proposition 7, $\boldsymbol{d}^{[\overline{p}]}(\boldsymbol{x})$ are Koopman-invariant coordinates of $\mathcal{G}_{\mathcal{K}_{\boldsymbol{f}}}$ evolving linearly with $\boldsymbol{A}_{[p]}(\tilde{\boldsymbol{A}})$ - concluding the proof. ■

A.2. Parameterizing Stable System Matrices

Proof of Lemma 8 To show the parameterization covers all Hurwitz matrices, we prove necessity and sufficiency. $\Rightarrow$: Let $\boldsymbol{X} = \boldsymbol{N}\boldsymbol{N}^\top + \epsilon\boldsymbol{I}$, $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{Q}^\top + \epsilon\boldsymbol{I}$ and $\boldsymbol{Z} = \frac{1}{2}\left(\boldsymbol{R} - \boldsymbol{R}^\top\right)$. We want to show that $\boldsymbol{A} = \boldsymbol{X}^{-1}\left(-\boldsymbol{Y} + \boldsymbol{Z}\right)$ is Hurwitz.
With $\boldsymbol{X}$ and $\boldsymbol{Y}$ symmetric real positive definite matrices we have $\boldsymbol{v}^\top\boldsymbol{Y}\boldsymbol{v} > 0\ \forall\boldsymbol{v} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$.
Since $\boldsymbol{Z}$ is a skew-symmetric real matrix, its eigenvalues are zero and $\boldsymbol{v}^\top\boldsymbol{Z}\boldsymbol{v} = 0$.
Now $\boldsymbol{v}^\top\left(-\boldsymbol{Y} + \boldsymbol{Z}\right)\boldsymbol{v} = -\boldsymbol{v}^\top\boldsymbol{Y}\boldsymbol{v} < 0$ and since $\boldsymbol{v}^\top\boldsymbol{X}^{-1}\boldsymbol{v} > 0$ finally

$$\boldsymbol{v}^\top\boldsymbol{A}\boldsymbol{v} = \boldsymbol{v}^\top\boldsymbol{X}^{-1}\left(-\boldsymbol{Y} + \boldsymbol{Z}\right)\boldsymbol{v} = \frac{\left(\boldsymbol{v}^\top\boldsymbol{X}^{-1}\boldsymbol{v}\right)\left(\boldsymbol{v}^\top\left(-\boldsymbol{Y} + \boldsymbol{Z}\right)\boldsymbol{v}\right)}{\boldsymbol{v}\boldsymbol{v}^\top} < 0$$

as $\boldsymbol{v}\boldsymbol{v}^\top > 0$.
$\Leftarrow$: From equation (13), $\left(\boldsymbol{N}\boldsymbol{N}^\top + \epsilon\boldsymbol{I}\right)\boldsymbol{A} + \epsilon\boldsymbol{I} = -\boldsymbol{Q}\boldsymbol{Q}^\top + \frac{1}{2}\left(\boldsymbol{R} - \boldsymbol{R}^\top\right)$ immediately follows. We use the facts that all real matrices can be decomposed into a symmetric and skew-symmetric part, and that all real symmetric positive definite matrices can be decomposed into $\boldsymbol{Q}\boldsymbol{Q}^\top$ via the Cholesky decomposition. Since $\boldsymbol{v}^\top\left(\boldsymbol{N}\boldsymbol{N}^\top + \epsilon\boldsymbol{I}\right)\boldsymbol{A}\boldsymbol{v} < 0$ for ϵ small enough, $\boldsymbol{Q}$ and $\boldsymbol{R}$ exist for all $\boldsymbol{N}$, decomposing $\left(\boldsymbol{N}\boldsymbol{N}^\top + \epsilon\boldsymbol{I}\right)\boldsymbol{A} + \epsilon\boldsymbol{I}$ into a (negative

definite) symmetric part $-\boldsymbol{Q}\boldsymbol{Q}^\top$ via the Cholesky decomposition and the corresponding skew-symmetric part $\frac{1}{2}\left(\boldsymbol{R}-\boldsymbol{R}^\top\right)$. ■

Proof of Theorem 9 By Lemma 8, the low-rank matrix $\tilde{\boldsymbol{A}}$ is Hurwitz by construction. Following Proposition 7, the eigenvalues of $\boldsymbol{A}_{[\bar{p}]}(\tilde{\boldsymbol{A}})$ satisfy $\sum_{i=1}^{d} \alpha_i \tilde{\lambda}_i$ with $\alpha_i \in \mathbb{N}_0$, making $\boldsymbol{A}_{[\bar{p}]}(\tilde{\boldsymbol{A}})$ Hurwitz as well - allowing us to replace the constraint (12b) with a Hurwitz matrix parameterization from Lemma 8. ■

A.3. Structured Relaxation of Exact Smooth Equivalence

Lemma 11 *Affine Coupling Flows with continuously differentiable scaling and translation mappings are diffeomorphisms.*

Proof of Lemma 11 Affine Coupling Flows are compositions of bijective coupling layers with the following forward

$$\mathbf{x}_a, \mathbf{x}_b = \mathrm{split}(\mathbf{x}) \tag{37a}$$
$$(\log \mathbf{s}, \mathbf{t}) = \mathrm{FA}\left(\mathbf{x}_b\right) \tag{37b}$$
$$\mathbf{s} = \exp(\log \mathbf{s}) \tag{37c}$$
$$\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t} \tag{37d}$$
$$\mathbf{y}_b = \mathbf{x}_b \tag{37e}$$
$$\mathbf{y} = \mathrm{concat}\left(\mathbf{y}_a, \mathbf{y}_b\right) \tag{37f}$$

and backward

$$\mathbf{y}_a, \mathbf{y}_b = \mathrm{split}(\mathbf{y}) \tag{38a}$$
$$(\log \mathbf{s}, \mathbf{t}) = \mathrm{FA}\left(\mathbf{y}_b\right) \tag{38b}$$
$$\mathbf{s} = \exp(\log \mathbf{s}) \tag{38c}$$
$$\mathbf{x}_a = \left(\mathbf{y}_a - \mathbf{t}\right)/\mathbf{s} \tag{38d}$$
$$\mathbf{x}_b = \mathbf{y}_b \tag{38e}$$
$$\mathbf{x} = \mathrm{concat}\left(\mathbf{x}_a, \mathbf{x}_b\right) \tag{38f}$$

structure (Kingma and Ba, 2015). Given ACFs are compositions of coupling layers, to prove ACFs are diffeomorphic maps we need the show the coupling layers are themselves diffeomorphisms - satisfying the following conditions:

[1] Bijective,

[2] Forward and inverse map C^1.

[1] Due to the special structure of (37) and (38), Coupling layers are bijective by construction. We can then consider the regularity of the bijection independently.

[2] Consider the function approximators employed in (37b) and (38b) are at least continuously differentiable[9]. The previous together with all other transformations in (37)

[9] This is not a strong assumption and covers many function approximators, e.g. kernel methods with $C^{\geq 1}$ kernels or neural networks with $C^{\geq 1}$ activation functions.

and (38) being arithmetic operations present arithmetic operations forward and backward maps preserve the regularity of the function approximator - as C^1-functions are closed under composition and arithmetic operations.

■

Proof of Theorem 10 With $\tilde{\boldsymbol{A}}$ Hurwitz by construction due to Lemma 8, following Proposition 7, the eigenvalues of $\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})$ are linear combinations $\sum_{i=1}^{d} \alpha_i \tilde{\lambda}_i$ of multi-index entries $\alpha_i \in \mathbb{N}_0$, making $\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})$ Hurwitz as well. By Proposition 7 the map $\boldsymbol{\varrho} : \boldsymbol{y} \mapsto \boldsymbol{y}^{[\overline{p}]}$, representing a monomial basis of the argument, is an immersion as $\operatorname{rank}(\boldsymbol{J}_{\boldsymbol{\varrho}}(\boldsymbol{y})) = \dim(\boldsymbol{y})$. Following Lemma 11, $\boldsymbol{d}$ is an immersion by construction. Its composition with the lifting map $\boldsymbol{\varrho} \circ \boldsymbol{d} : \boldsymbol{y} \mapsto \boldsymbol{d}^{[\overline{p}]}$ is as well due to immersions being invariant under composition. With $\boldsymbol{A}_{[\overline{p}]}(\tilde{\boldsymbol{A}})$ Hurwitz and $\boldsymbol{d}^{[\overline{p}]}$ immersible, the asymptotic stability of the lifted model (6a)-(6b) follows via (Yi and Manchester, 2021, Propositon 1). Hence, every optimization problem (15) yields an asymptotically stable system. ■

Appendix B. Details on the comparison versus SKEL Fan et al. (2021)

B.1. Parameters and Implementation

The KF-SDS algorithm was implemented in PyTorch[10]. For all experiments, ACF with 10 coupling layers are used to learn the diffeomorphisms. The neural networks for the scaling and translation functions in each affine coupling layer have 3 hidden layers, with 120 neurons, each with a smooth Exponential Linear Unit (ELU) as the activation function. The dimension of the lifting coordinates is $D = 44$ ($\bar{p} = 8$). We employ the ADAM optimizer with a learning rate of 0.0025 and full batch training. Further we use weight decay and clip gradients. We do not use weights on loss terms, due to the highly coupled structure of the optimization problem (14). To ease the training process we initialize KF-SDS with the least-square linear model: the diffeomorphic system matrix $\tilde{\boldsymbol{A}}$ with $\tilde{\boldsymbol{A}} = \arg\min \|\dot{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{x}\|_2^2$, the ACF to be the identity map and the reconstruction matrix $\boldsymbol{C}$ to reconstruct the linear diffeomorphic system. The aggregate system- and lift-equations are obtained symbolically. Our models used for comparison were trained for 5000 epochs.

The SKEL (Fan et al., 2021) models were trained as in Fan et al. (2021), but on all seven demonstrations: They have a concatenated immediate state $\boldsymbol{z} = [\boldsymbol{x}^\top, \dot{\boldsymbol{x}}^\top]^\top$, a 20-dimensional lifted system matrix and use fully-connected-neural-networks with two hidden-layers with 50 nodes each for nonlinear lifting and reconstruction maps. They were trained for 50000 epochs with a batch-size of 1000.

To ensure a fair comparison to SKEL we use the same base data to train both models. Due to the the discrete time nature of SKEL the LASA-trajectories are resampled with $dt = 0.05s$. As training data for KF-SDS we sampled 900 datapoints per demonstration from the SKEL dataset, using cubic interpolation. This ensures that both models see a similar amount of datapoints (with repetition) during the training process. For both methods data is normalized to stay within the unit box. Notably both SKEL and KF-SDS, although being discrete-time and continuous-time, use the same input data: positions and velocities.

[10] https://github.com/pytorch/pytorch

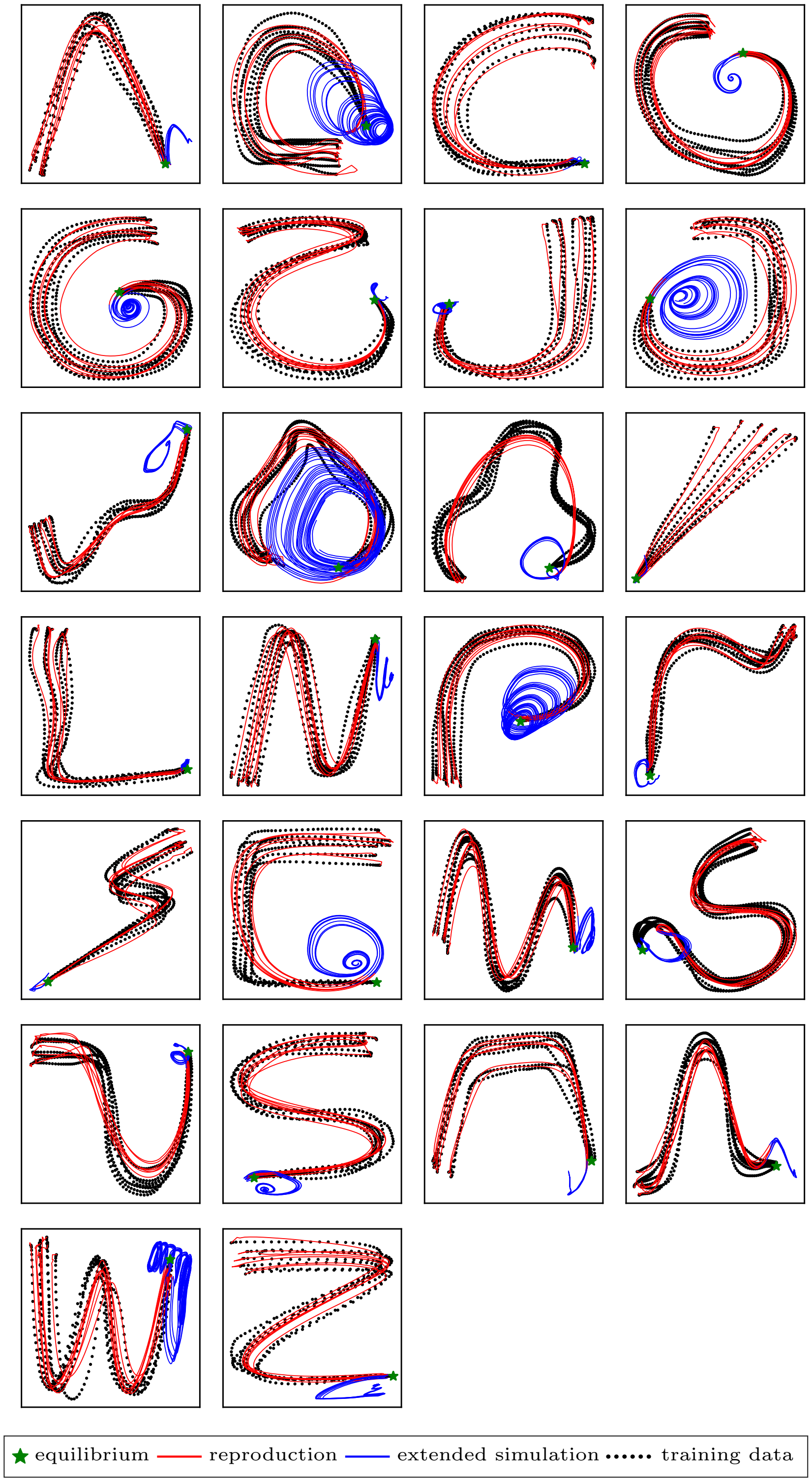

Figure 4: SKEL trajectories until five times the demonstration end-time

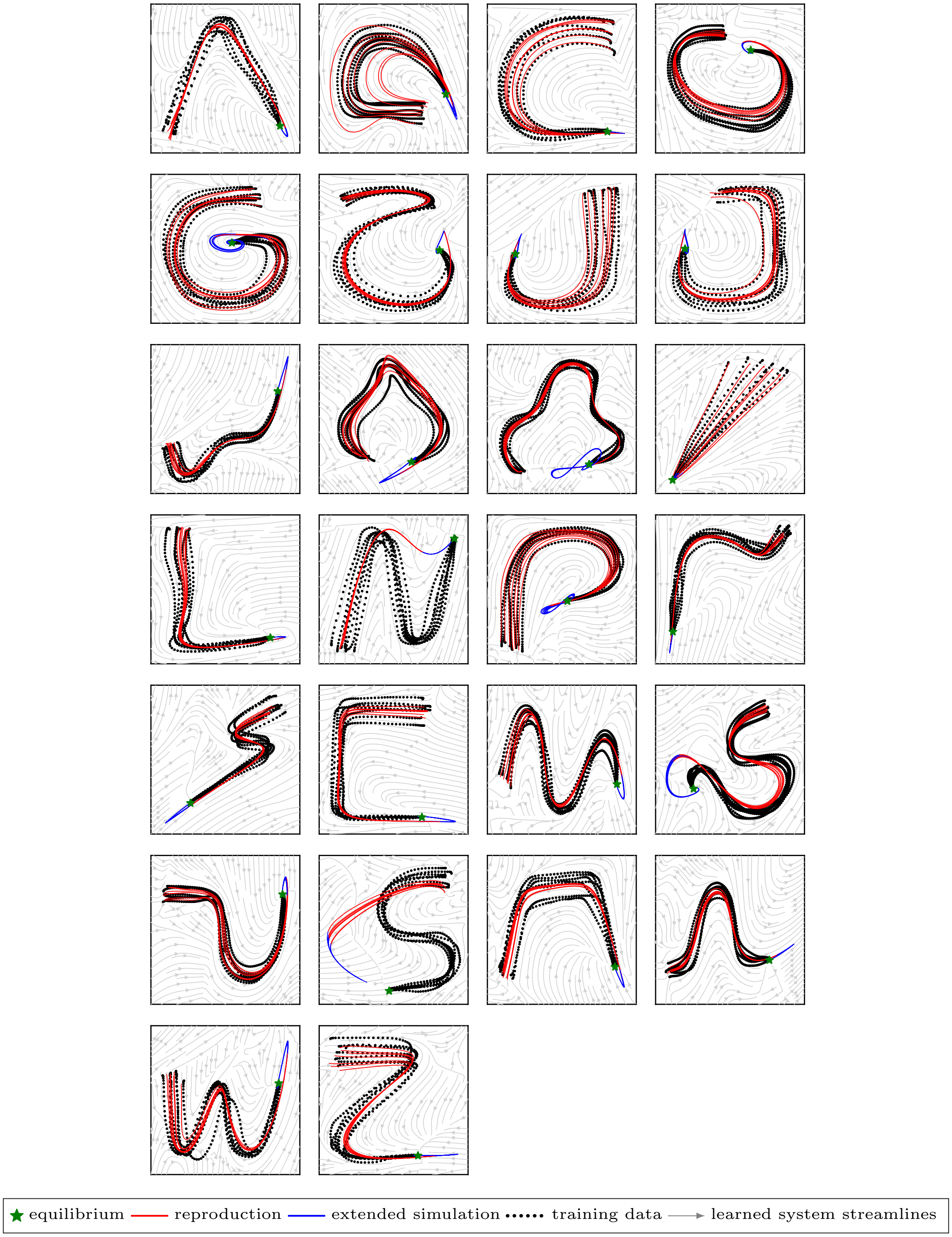

Figure 5: KF-SDS trajectories until five times the demonstration end-time

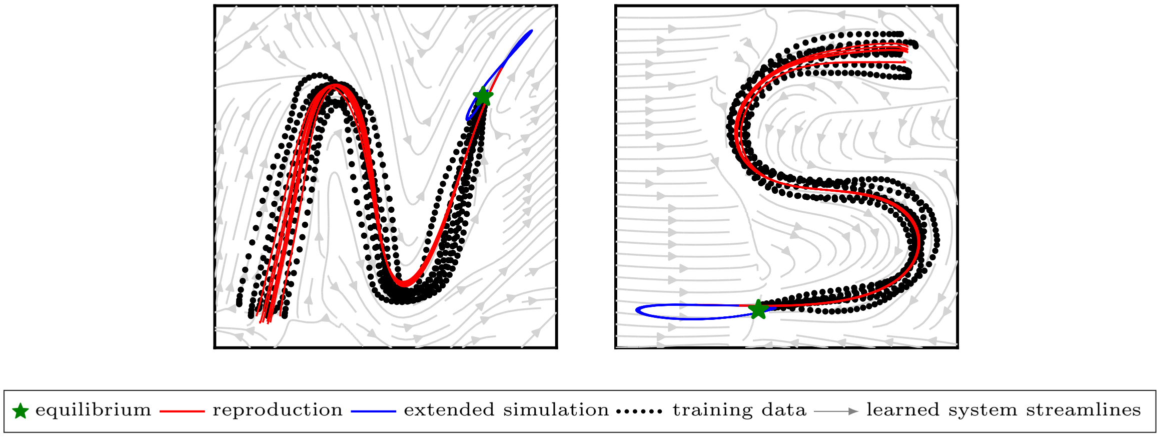

Figure 6: KF-SDS trained for 20000 epochs

B.2. Discussion

Figure 4 and 5 show the simulated trajectories until the demonstration end-time (red) and its continuation (blue) until five times the demonstration time for SKEL (Fan et al., 2021) and KF-SDS, respectively. While SKEL manages to reproduce the demonstrations, it fails at guaranteeing long term behavior due to its nonlinear reconstruction. Long-term behavior is a major advantage of KF-SDS, as its linear reconstruction with the special construction of lifting features ensures well behaved convergence to the equilibrium. The linear evolution leads to a settling phase in the neighborhood of the equilibrium. Note that both models are ODEs in the lifted and aggregated space respectively, this does not necessarily apply to the immediate state-space, allowing for sharp and self-intersecting, but slow, immediate state trajectories near the equilibrium. For physical dynamical systems, like human handwriting motions, a settling phase is to be expected since velocities will approach zero in a rather smooth way.

For shapes N and S, KF-SDS did not converge to a satisfactory solution with the prescribed 5000 epochs. This is, however, due to slower convergence, rather than convergence to a local minimum and is resolved by extended training as shown in Figure 6. Our results show how KF-SDS can predict system dynamics in a linear fashion for a long time horizon given the duration of the demonstrations ranges from 1.9 to 8.7 seconds. This makes it promising for motion generation and model simplification, practicable for receding horizon prediction and control strategies.