Machine Learning Enhanced Hankel Dynamic-Mode Decomposition

Christopher W. Curtis[1], D. Jay Alford-Lago[1,2], Erik Bollt[3], and Andrew Tuma[1]

[1]*Department of Mathematics and Statistics, San Diego State University*
[2]*Naval Information Warfare Center Pacific*
[3]*Department of Mathematics, Clarkson University*

Abstract

While the acquisition of time series has become increasingly more straightforward and sophisticated, developing dynamical models from time series is still a challenging and ever evolving problem domain. Within the last several years, to address this problem, there has been a merging of machine learning tools with what is called the dynamic mode decomposition (DMD). This general approach has been shown to be an especially promising avenue for sophisticated and accurate model development. Building on this prior body of work, we develop a deep learning DMD based method which makes use of the fundamental insight of Takens' Embedding Theorem to develop an adaptive learning scheme that better captures higher dimensional and chaotic dynamics. We call this method the Deep Learning Hankel DMD (DLHDMD). We show that the DLHDMD is able to generate accurate dynamics for chaotic time series, and we likewise explore how our method learns mappings which tend, after successful training, to significantly change the mutual information between dimensions in the dynamics. This appears to be a key feature in enhancing the DMD overall, and it should help provide further insight for developing more sophisticated deep learning methods for time series forecasting.

1 Introduction

The incorporation of modern machine learning methodology into dynamical systems is creating an ever expanding array of techniques pushing the boundaries of what is possible with regards to describing nonlinear multi-dimensional time series. Longstanding problems such as finding optimal Takens' embeddings [1, 2] now have powerful and novel deep learning based

algorithmic approaches [3] which would not have been feasible even ten years ago. Likewise, the field of equation free modeling using Koopman operator methods, broadly described by Dynamic Mode Decomposition (DMD), has seen several innovative deep learning based methods emerge over the last several years [4, 5, 6] which have been shown to greatly expand the accuracy and flexibility of DMD based approaches. There have also been related and significant advances in model identification and solving nonlinear partial differential equations via deep learning techniques [7, 8, 9, 10].

With this background in mind, in this work we focus on extending the methods in [6] which were called Deep Learning DMD (DLDMD). In that work, a relatively straightforward method merging auto-encoders with the extended DMD (EDMD) was developed. This was done by using an encoder to embed dynamics in a sufficiently high enough dimensional space which then generated a sufficiently large enough space of observables for the EDMD to generate accurate linear models of the embedded dynamics. Decoding then returned the embedded time series to the original variables in such a way as to guarantee the global stability of iterating the linear model to generate both reconstructions and forecasts of the dynamics. The DLDMD was shown to be very effective in finding equation-free models which were able to both reconstruct and then forecast from data coming from planar dynamical systems.

However, when chaotic time series from the Lorenz-63 system were examined, the performance of the DLDMD was found to degrade. While this clearly makes the DLDMD approach limited in its scope, we note that the successful use of DMD based approaches to accurately reconstruct or forecast chaotic dynamics are not readily available. Other methods such as HAVOK [11] or SINDy [12] are more focused on the analysis of chaotic time series or the discovery of model equations which generate chaotic dynamics, though of course if one has an accurate model, then one should be able to generate accurate forecasts. In this vein, there are also methods using reservoir computing (RC) [13, 14], though again, nonlinear models are essentially first learned and then used to generate forecasts. However, both SINDy and RC rely on proposing libraries of terms to build models which are then fit (or learned from) data.

While effective, such approaches do not allow for the spectral or modal analysis which has proven to be such an attractive and useful feature of DMD based methods. Likewise, they require a number of user decisions about how to construct the analytic models used in later regressive fitting that amount to a guess and check approach to generating accurate reconstructions and forecasts. Therefore in this work, using insights coming from the Takens' Embedding Theorem (TET) [15, 3], we expand the DLDMD framework so as to make it accurate in both generating reconstructions and forecasts of chaotic time series. This is done by first making the EDMD over embedded coordinates global as opposed to the local approach of [6]; see also [4, 5]. Sec-

ond, we develop an adaptive Hankel matrix based ordering of the embedded coordinates which adds more expressive power for approximating dynamics to the deep learning framework. To study our method, we use data generated by the Lorenz-63 and Rossler systems as well as twelve-dimensional projections of data from the Kuramoto–Sivashinksky (KS) equation. In all of these cases, we show that by combining our proposed modifications to the DLDMD that we are able to generate far more accurate reconstructions for chaotic systems than with DLDMD alone. Moreover, we have built a method which still allows for the straightforward modal analysis which DMD affords and keeps user choices to a handful of real-valued hyperparamters while still producing results competitive with other approaches in the literature.

Further, motivated by the classic information theory (IT) studies of the TET [1], as well as modern insights into the role that information plays in deep learning [16, 17], we study how the fully trained encoder changes the information content of the dynamics coming from the Lorenz-63 and Rossler systems. For the Lorenz-63 system, the encoder tends to either slightly decrease the mutual information or cause strong phase shifts which decrease the coupling times across dimensions. However, the characteristic timescales corresponding to lobe switching in the Lorenz 'butterfly' are clearly seen to be preserved in the dynamics of the information for the Lorenz-63 system. In contrast then, for the Rossler system, the slow/fast dichotomy in the dynamics seen in the original coordinates is made more uniform so that rapid transients in the information coupling are removed by the encoder. Thus in either case, we see that the encoder generates significant differences in the information content between dimensions in the latent coordinates relative to the original ones, and that this strong change in information content is a critical feature in successful training.

Of course, the present work is ultimately preliminary, and there are a number of important questions left to be resolved. First, while we are able to easily display computed spectra, the affiliated global Koopman modes we find are not as straightforward to show. We generate our results from random initial conditions, so the most effective means of constructing the global Koopman modes would be via radial-basis functions, but the implementation would be nontrivial due to the infamous ill-conditioning issues which can plague the approach [18]. Second, there is a clear need for a comparison across SINDy, RC, and our DLHDMD methods. In particular, the present work generates excellent reconstructions and thus modal decompositions, but learning a method which generates accurate longer time novel predictions beyond the given data has proven too challenging thus far. How well other methods address this issue relative to their reconstruction and other diagnostic properties, and then how all of these methods compare in these several different ways is as yet unclear. While acknowledging then the limitations of the present work, we defer addressing the above issues till later works where each of the above issues can be dealt with in the detail

that is needed.

The structure of this paper is as follows. In Section 2, we provide an introduction to the Extended DMD and then explain the extensions we develop which are critical to the success of the present work. In Section 3, we introduce the Hankel DMD and, incorporating the extensions introduced in Section 2, we show how well it does and does not perform on several examples. Then in Section 4 we introduce the Deep Learning Hankel DMD and provide results on its performance as well as an analysis of how the mutual information changes in the latent variables. Section 5 presents our results on mutual information. Section 6 provides conclusion and discussion.

2 Extended Dynamic Mode Decomposition

To begin, we suppose that we have the data set $\{\mathbf{y}_j\}_{j=1}^{N_T+1}$ where

$$\mathbf{y}_j = \varphi(t_j; \mathbf{x}),\ t_{j+1} = t_j + \delta t,\ \mathbf{x} \in \mathbb{R}^{N_s}$$

where δt is the time step at which data is sampled and $\varphi(t; \mathbf{x})$ is a flow map such that $\varphi(t_1, \mathbf{x}) = \mathbf{x}$. From the flow map, we define the affiliated *Koopman operator* $\mathcal{K}^t$ such that for a given scalar observable $g(\mathbf{x})$, one has

$$\mathcal{K}^t g(\mathbf{x}) = g(\varphi(t, \mathbf{x})),$$

so that the Koopman operator linearly tracks the evolution of the observable along the flow. We likewise define the associated Hilbert space of *observables*, say $L_2\left(\mathbb{R}^{N_s}, \mathbb{R}, \mu\right)$, or more tersely as $L_2(\mathcal{O})$, so that $g \in L_2(\mathcal{O})$ if

$$\int_{\mathbb{R}^{N_s}} |g(\mathbf{x})|^2 \, d\mu(\mathbf{x}) < \infty,$$

where μ is some appropriately chosen measure. This makes the infinite-dimensional Koopman operator $\mathcal{K}^t$ a map such that $\mathcal{K}^t : L_2(\mathcal{O}) \to L_2(\mathcal{O})$.

Following [19, 20], given our time snapshots $\{\mathbf{y}_j\}_{j=1}^{N_T+1}$, we suppose that any observable $g(\mathbf{x})$ of interest lives in a finite-dimensional subspace $\mathcal{F}_D \subset L_2(\mathcal{O})$ described by a given basis of observables $\{\psi_l\}_{l=1}^{N_{ob}}$ so that

$$g(\mathbf{x}) = \sum_{l=1}^{N_{ob}} a_l \psi_l(\mathbf{x}).$$

Given this ansatz, we then suppose that

$$\begin{aligned}\mathcal{K}^{\delta t} g(\mathbf{x}) &= \sum_{l=1}^{N_{ob}} a_l \psi_l(\varphi(\delta t, \mathbf{x})) \\ &= \sum_{l=1}^{N_{ob}} \psi_l(\mathbf{x}) \left(\mathbf{K}_a^T \mathbf{a}\right)_l + r(\mathbf{x}; \mathbf{K}_a)\end{aligned}$$

where $r(\mathbf{x};\mathbf{K}_a)$ is the associated error which results from the introduction of the finite-dimensional approximation of the Koopman operator represented by $\mathbf{K}_a$. We can then find $\mathbf{K}_a$ by solving the following minimization problem

$$\begin{aligned}\mathbf{K}_a =& \arg\min_{\mathbf{K}} |r(\mathbf{x};\mathbf{K})|^2 \\ =& \arg\min_{\mathbf{K}} \sum_{j=1}^{N_T} \left| \sum_{l=1}^{N_{ob}} \left(a_l \psi_l(\mathbf{y}_{j+1}) - \psi_l(\mathbf{y}_j) \left(\mathbf{K}^T \mathbf{a}\right)_l \right) \right|^2 \\ =& \arg\min_{\mathbf{K}} \sum_{j=1}^{N_T} |\langle \mathbf{\Psi}_{j+1} - \mathbf{K}\mathbf{\Psi}_j, \mathbf{a}^* \rangle|^2 ,\end{aligned} \tag{1}$$

where $\mathbf{a} = (a_1 \cdots a_{N_{ob}})^T$, $\mathbf{\Psi}_j = (\psi_1(\mathbf{y}_j) \cdots \psi_{N_{ob}}(\mathbf{y}_j))^T$, the inner product $\langle , \rangle$ is the standard one over $\mathbb{C}^{N_{ob}}$, and the $*$ symbol denotes complex conjugation. It is straightforward to show that an equivalent and easier to solve form of this optimization problem is given by

$$\mathbf{K}_a = \underset{\mathbf{K}}{\operatorname{argmin}} ||\mathbf{\Psi}_+ - \mathbf{K}\mathbf{\Psi}_-||_F^2 , \tag{2}$$

where $||\cdot||_F$ is the Frobenius norm, and the $N_{ob} \times N_T$ matrices $\mathbf{\Psi}_\pm$ are given by

$$\mathbf{\Psi}_- = \{\mathbf{\Psi}_1 \ \mathbf{\Psi}_2 \ \cdots \ \mathbf{\Psi}_{N_T}\}, \quad \mathbf{\Psi}_+ = \{\mathbf{\Psi}_2 \ \mathbf{\Psi}_3 \ \cdots \ \mathbf{\Psi}_{N_T+1}\}.$$

In practice, we solve this equation using the Singular-Value Decomposition (SVD) of $\mathbf{\Psi}_-$ so that

$$\mathbf{\Psi}_- = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^\dagger.$$

This then gives us

$$\mathbf{K}_a = \mathbf{\Psi}_+ \mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{U}^\dagger,$$

with the corresponding error in the Frobenius norm $E_r(\mathbf{K}_a)$ where

$$E_r(\mathbf{K}_a) = \left|\left| \mathbf{\Psi}_+ \left(I - \mathbf{W}\mathbf{W}^\dagger \right) \right|\right|_F .$$

To complete the algorithm, after diagonalizing $\mathbf{K}_a$ so that

$$\mathbf{K}_a = \mathbf{V}\mathbf{L}\mathbf{V}^{-1}, \ \mathbf{L}_{ll} = \ell_l, \tag{3}$$

then one can show that the Koopman eigenfunctions $\phi_l(\mathbf{y}_j)$ are found via the equations

$$\mathbf{\Phi}_\pm = \mathbf{V}^{-1}\mathbf{\Psi}_\pm. \tag{4}$$

From here, one can, starting from the initial conditions, approximate the dynamics via the reconstruction formula

$$\mathbf{y}(t;\mathbf{x}) \approx \sum_{l=1}^{N_{ob}} \mathbf{k}_l e^{t\lambda_l} \phi_l(\mathbf{x}), \tag{5}$$

where $\lambda_l = \ln(\ell_l)/\delta t$ and the *Koopman modes* $\mathbf{k}_l \in \mathbb{C}^{N_s}$ solve the initial-value problem

$$\mathbf{x} = \sum_{l=1}^{N_{ob}} \mathbf{k}_l \phi_l(\mathbf{x}).$$

Again, in matrix/vector notation, keeping in mind that $\mathbf{x} \in \mathbb{R}^{N_s}$ and that in general $N_s \neq N_{ob}$, we have

$$\mathbf{x} = \mathbf{K}_M \begin{pmatrix} \phi_1(\mathbf{x}) \\ \vdots \\ \phi_{N_{ob}}(\mathbf{x}) \end{pmatrix}$$

where $\mathbf{K}_M$ is the $N_s \times N_{ob}$ matrix whose columns are the Koopman modes $\mathbf{k}_j$. As can be seen then, generically, one can only find the Koopman modes through least-squares solutions of the non-square problem. In this regard, one would do well to have information from as many initial conditions as possible to over-determine the problem.

2.1 Extensions to EDMD

To wit, if we had a collection of initial conditions $\{\mathbf{x}_k\}_{k=1}^{N_C}$ with corresponding path data $\{\mathbf{y}_{j,k}\}_{j,k=1}^{N_T+1,N_C}$, we can extend the optimization problem in Equation (1) to be

$$\mathbf{K}_a = \arg\min_{\mathbf{K}} \sum_{k=1}^{N_C} |r(\mathbf{x}_k; \mathbf{K})|^2 ,$$

so that now the problem of finding $\mathbf{K}_a$ is no longer strictly localized to a particular path labeled by the initial condition $\mathbf{x}$. Following the same logic above leads one to simply concatenate across observables column wise when generating the $\mathbf{\Psi}_\pm$ matrices so that

$$\mathbf{\Psi}_- = \{\mathbf{\Psi}_{1,1}\ \mathbf{\Psi}_{2,1}\ \cdots\ \mathbf{\Psi}_{N_T,1}\ \cdots \mathbf{\Psi}_{1,N_C}\ \mathbf{\Psi}_{2,,N_C}\ \cdots\ \mathbf{\Psi}_{N_T,N_C}\}$$

where

$$\mathbf{\Psi}_{j,k} = (\psi_1(\varphi(t_j; \mathbf{x}_k)) \cdots \psi_{N_{ob}}(\varphi(t_j; \mathbf{x}_k)))^T$$

The matrix $\mathbf{\Psi}_+$ is defined similarly. Using then the EDMD algorithm outlined above, we arrive at the following matrix problem for determining $\mathbf{K}_M$

$$\mathbf{X} = \mathbf{K}_M \mathbf{\Phi}_0$$

where

$$\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_{N_C}),\ \mathbf{\Phi}_0 = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_1(\mathbf{x}_{N_C}) \\ \phi_2(\mathbf{x}_1) & \cdots & \phi_2(\mathbf{x}_{N_C}) \\ \vdots & \vdots & \vdots \\ \phi_{N_{ob}}(\mathbf{x}_1) & \cdots & \phi_{N_{ob}}(\mathbf{x}_{N_C}). \end{pmatrix}$$

Likewise, given that Equation (4) gives us time series of the Koopman eigenfunctions, which necessarily must satisfy, assuming sufficient accuracy of the approximation implied by Equation (3), the identity

$$\phi_l\left(\varphi(t_j;\mathbf{x}_k)\right) = \mathcal{K}^j\phi_l(\mathbf{x}_k) = \ell_l^j\phi_l\left(\mathbf{x}_k\right),$$

we can generalize Equation (5) via the model

$$\mathbf{Y}_{N_{st}} \approx \mathbf{K}_M\mathbf{L}^{N_{st}}\mathbf{\Phi}_-,\ N_{st}\in\mathbb{N}\cup\{0\}, \tag{6}$$

where

$$\mathbf{Y}_{N_{st}} \approx \left\{\mathbf{y}_{N_{st},1}\cdots\mathbf{y}_{N_{st}+N_T,1}\ \cdots\ \mathbf{y}_{N_{st},N_C}\cdots\mathbf{y}_{N_{st}+N_T,N_C}\right\},$$

which generates a reconstruction of the data for time steps $N_{st}\leq j\leq N_T+1$ and a forecast for steps with index $N_T+1\leq j\leq N_T+N_{st}$. Using this formula allows for far greater flexibility in employing the EDMD since we can control how many times steps for which we wish to generate reconstructions, which is relatively easy. This is in contrast to generating forecasts through the iteration of the diagonal matrix $\mathbf{L}$, which is a process that is generally sensitive to small variations in the position of the eigenvalues ℓ_l, especially for those near the unit circle in the complex plane. We will make great use of this generalization in the later sections of this paper.

3 Hankel DMD

When implementing EDMD, the most natural observables are the projections along the canonical Cartesian axes, i.e.

$$\psi_l(\mathbf{x}) = x_l,\ l=1,\cdots,N_s.$$

If we stick to this space of observables, the EDMD method reduces to the standard DMD method. Thus the idea with EDMD is to include more nonlinear observables to hopefully represent a richer subspace of dynamics and thereby make the approximation of corresponding Koopman operator more accurate and sophisticated.

With this in mind, [15] built upon the classic idea of Takens embeddings [21] and explored using affiliated Hankel matrices to generate natural spaces of observables for EDMD, and approach we describe as Hankel DMD (HDMD). Also of note in this direction is the HAVOK method developed in [11], though in some ways HAVOK is more akin to the *embedology* methods explored in such classic works as [22, 2].

HDMD thus begins with an affiliated scalar measurement of our time series, say $\{g(\mathbf{y}_j)\}_{j=1}^{N_T+1}$. From this, by introducing a *window* size N_w one

builds the affiliated Hankel matrix $\tilde{\mathbf{H}}_g(\mathbf{x})$ where

$$\tilde{\mathbf{H}}_g(\mathbf{x}) = \begin{pmatrix} g(\mathbf{y}_1) & \cdots & g(\mathbf{y}_{N_w}) \\ g(\mathbf{y}_2) & \cdots & g(\mathbf{y}_{N_{w+1}}) \\ \vdots & \vdots & \vdots \\ g(\mathbf{y}_{N_{ob}}) & \cdots & g(\mathbf{y}_{N_T+1}) \end{pmatrix}.$$

where the number of observables $N_{ob} = N_T + 1 - (N_w - 1)$.

What one sees then is that each row of $\tilde{H}_f(\mathbf{x})$ is some iteration of the Koopman operator $\mathcal{K}^{\delta t}$. From here then, each row of N_w time steps is defined to be its own separate observable $\psi_l(\mathbf{x})$, i.e.

$$\psi_l(\mathbf{x}) = \mathcal{K}^{l\delta t} g(\mathbf{x}),\ l = 1, \cdots, N_{ob}.$$

One then proceeds as above with the EDMD algorithm, where we emphasize that N_T is replaced by $N_w - 1$. This is an interesting feature, or arguably limitation, of the HDMD method in which we generate matrices $\Phi_\pm$ (see Equation (4)) up to the time index $N_w - 1 \leq N_T$. Thus later times are used to build approximations at prior times. This makes the issue of forecasting data more difficult since one must iterate the EDMD results, as is done via Equation (6), from time index $N_w - 1$ up to N_T to reconstruct the original data that was used in the first place. Throughout the remainder of the paper then, we take care to distinguish between *iterated reconstructions* and actual *forecasts* which make novel predictions beyond the given data.

Finishing our explanation of HDMD, if one has data along multiple initial conditions, say $\{\mathbf{x}_k\}_{k=1}^{N_C}$, we can extend the above algorithm by concatenating Hankel matrices so that we perform EDMD on the combined matrix $\tilde{\mathbf{H}}_C$ so that

$$\tilde{\mathbf{H}}_C = \begin{pmatrix} \tilde{H}_g(\mathbf{x}_1) & \cdots & \tilde{H}_g(\mathbf{x}_{N_C}) \end{pmatrix}$$

The inclusion of other observables can be done in a similar fashion.

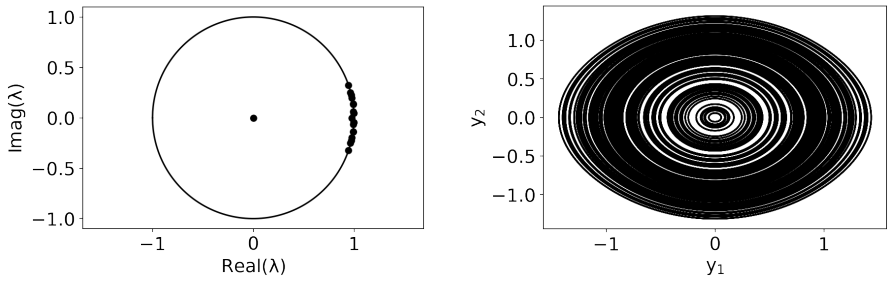
3.1 Results for HDMD

The ultimate promise of the HDMD is that it should facilitate an adaptable implementation of the EDMD framework which allows for the number of observables to simply be adjusted by the window size. To see this, in all of the following results we let $t_f = 20$, $dt = .05$, and we use $N_C = 128$ random initial conditions which are then stacked together. For HDMD, observables along each dimension of the dynamical system are used. Reconstructions and forecasts are generated using Equation (6) for $N_{st} = 20$, which for a time step of $dt = .05$, means forecasts are produced up to a unit of non-dimensional time. We note though that the choice of N_{ob} defines the variable $N_w = N_T + 1 - (N_{ob} - 1)$, so that instead of using EDMD on data from $0 \leq t \leq t_f$, we now use data from $0 \leq t \leq t_{f,w}$ where $t_{f,w} = N_w \delta t$.

If we then take data from the standard harmonic oscillator, where for $\mathbf{y}(t) = (y_1(t), y_2(t))^T$ we have

$$\dot{y}_1 = y_2,\ \dot{y}_2 = -\sin(y_1),\ \mathbf{y}(0) = \mathbf{x},$$

then HDMD produces the results seen in Figure 1. Using $N_{ob} = 10$, excellent iterated reconstructions and forecasts (note $N_{ob} < N_{st}$) are obtained for the entire field of initial conditions examined. The computed eigenvalues are largely localized along the complex unit circle. We emphasize that the HDMD method does this without any added guidance or control on the part of the user.



(a) (b)

Figure 1: HDMD results for the harmonic oscillator with $N_{ob} = 10$, so $t_{f,w} = 19.5$, and $N_{st} = 20$, so that the reconstruction is generated for times $1 \leq t \leq t_{f,w}$ and iterated reconstruction and forecasting is done for times $t_{f,w} \leq t \leq t_{f,w} + 1$. The computed eigenvalues using $N_{ob} = 10$ are shown in (a) and the trajectory reconstructions and forecasts are shown in (b).

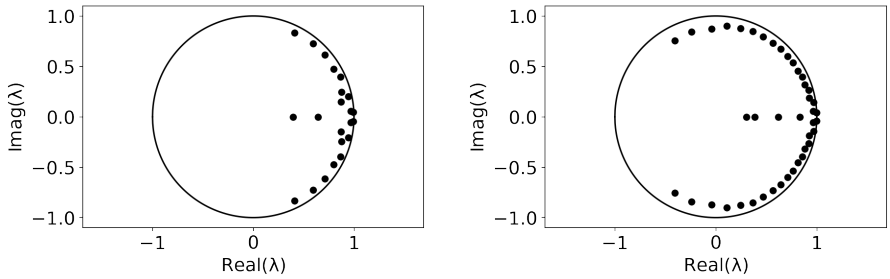
Moving on to the more complicated case of the Van der Pol oscillator, where

$$\dot{y}_1 = y_2,\ \dot{y}_2 = -y_1 + \mu(1 - y_1^2)y_2,\ \mu = 1.5,$$

we find, as seen in Figure 2, that $N_{ob} = 10$ does not produce as accurate of reconstructions and forecasts as we readily got for the harmonic oscillator. By increasing N_{ob} to 20 though, we are able to generate far more accurate results, though at the cost of being able to forecast beyond the given time series. We further note that by fixing $N_{ob} = 20$ and letting $N_{st} = 30$, we get essentially the same degree of degradation in the forecast as when we chose $N_{ob} = 10$ and $N_{st} = 20$. This limitation aside, we see in call cases that the eigenvalues generated in this method naturally fall on or inside the unit circle, thereby generating very stable, even if inaccurate, dynamics.

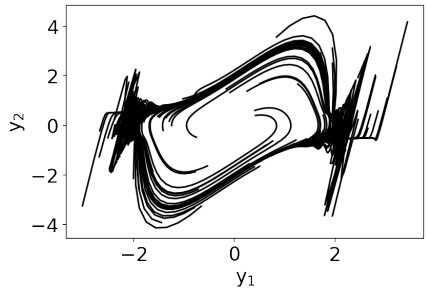
In contrast to these results, we find that the Lorenz Equations

$$\begin{aligned}
\dot{y}_1 =& \sigma(y_2 - y_1), \\
\dot{y}_2 =& \rho y_1 - y_2 - y_1 y_3, \\
\dot{y}_3 =& -by_3 + y_1 y_2,
\end{aligned} \tag{7}$$

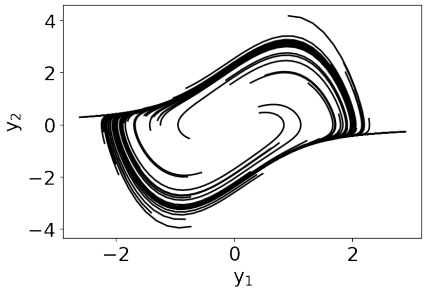


(a) $N_{ob} = 10$

(b) $N_{ob} = 20$



(c) $N_{ob} = 10$



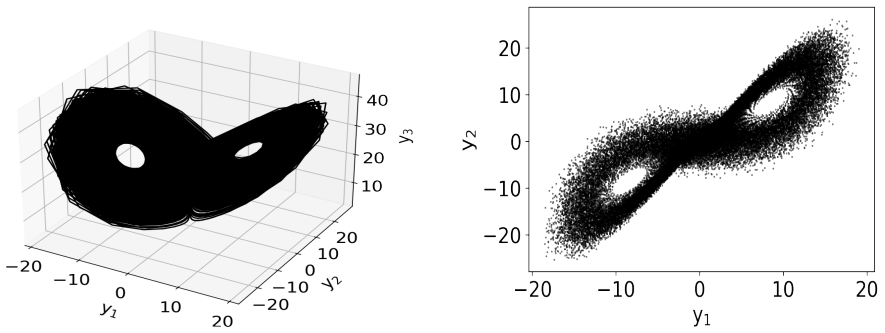
(d) $N_{ob} = 20$

Figure 2: HDMD results for the Van der Pol equation with $N_{st} = 20$ and $N_{obs} = 10$ and $N_{obs} = 20$, so that the reconstruction is generated for $1 \leq t \leq t_{f,w}$ and iterated reconstruction and forecasting is done for times $t_{f,w} \leq t \leq t_{f,w} + 1$ where $t_{f,w} = 19.5$ for $N_{obs} = 10$ and $t_{f,w} = 19$ for $N_{obs} = 20$. We note the enhanced accuracy for $N_{ob} = 20$ comes at the expense of generating novel forecasts of the time series. Along the top row are the eigenvalue plots, while reconstructions are presented along the bottom row. As can be seen, relative to the choice of N_{st}, doubling the number of observables greatly enhances the accuracy of the reconstructions and forecast.

where

$$\sigma = 10, \ \rho = 28, \ b = \frac{8}{3},$$

provide a case in which the HDMD is not able to adequately capture the dynamics for any reasonable choices of N_{ob}. This is not necessarily surprising given that for the parameter choices made, we know that the dynamics traces out the famous Butterfly strange attractor as seen in Figure 3. Given that we are trying to approximate dynamics on a strange attractor, we would reasonably anticipate the HDMD to struggle. However, as seen in Figure 4, we see the method essentially fails completely for parameter choices identical to those used above. Arguably, by comparing Figures 4 (e) and (f) to one another, we see that doubling the number of observables gives one a better approximation of the (y_1, y_2) projection of the Butterfly, but that is a coarse metric at best. That all said, the position of the computed spectra seen in



(a) (b)

Figure 3: The Lorenz Butterfly in (a) with its projection along the (y_1, y_2) plane in (b).

Figures 4 (a) and (b) is still relatively ideal, so further adaptation of the HDMD method might produce more desirable results. We will see how to realize this through the use of neural networks in the following section.

4 Deep Learning HDMD

To improve the HDMD such that it is able to deal with chaotic systems such as the Lorenz equation, we now turn to and adapt the framework of the deep learning DMD (DLDMD) developed in [6]. Our deep learning enhanced HDMD begins with an autoencoder composed of neural networks $\mathcal{E}$ (the encoder) and $\mathcal{D}$ (the decoder) such that

$$\mathcal{E} : \mathbb{R}^{N_s} \to \mathbb{R}^{N_s},\ \mathcal{D} : \mathbb{R}^{N_s} \to \mathbb{R}^{N_s},$$

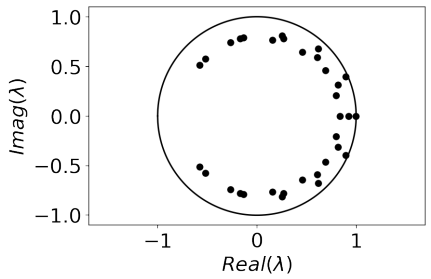
and such that our auto-encoder is a near identity, i.e.

$$\tilde{\mathbf{y}} = \mathcal{E}(\mathbf{y}),\ \mathcal{D}(\tilde{\mathbf{y}}) \approx \mathbf{y}.$$

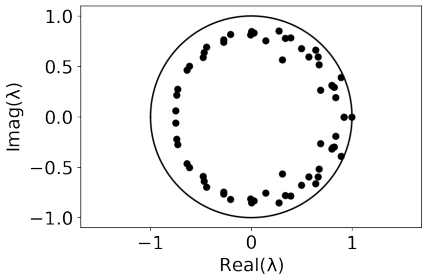
Note, we call the encoded coordinates *latent variables* or *latent dimensions* in line with the larger literature on machine learning.

The encoded coordinates should represent a set of observables which should enhance the overall accuracy of HDMD approximations of the dynamics. To train for this, after making reasonable choices for how to initialize the weights of the auto-encoder, and fixing a choice for N_{st}, given the training data, say $\{\mathbf{y}_{j,k}\}_{j,k=1}^{N_T+1,N_C}$, and the validation data $\left\{\mathbf{y}_{j,k}^{(vl)}\right\}_{j,k=1}^{N_T+1,N_C^{(vl)}}$, we use the following loss function

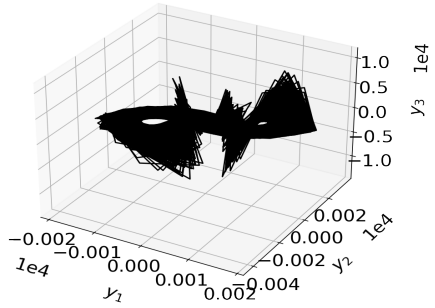
$$\mathcal{L}_{tot} = \mathcal{L}_{recon} + \mathcal{L}_{pred} + \mathcal{L}_{dmd} + \alpha\mathcal{L}_{reg}$$



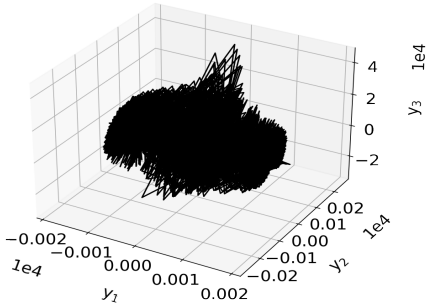
(a) $N_{ob} = 10$



(b) $N_{ob} = 20$



(c) $N_{ob} = 10$



(d) $N_{ob} = 20$

Figure 4: HDMD results for the Lorenz system with $N_{st} = 20$, so that the reconstruction is generated for times $1 \leq t \leq t_{f,w}$ and iterated reconstruction and forecasting is done for times $t_{f,w} \leq t \leq t_{f,w} + 1$ where $t_{f,w} = 19.5$ for $N_{ob} = 10$ and $t_{f,w} = 19$ for $N_{ob} = 20$. In the top row are eigenvalue plots, while reconstructions are presented along the bottom row. As can be seen, doubling the number of observables does little to enhance the accuracy of the reconstructions.

where

$$\mathcal{L}_{recon} = \left[\frac{1}{N_T + 1} \sum_{j=1}^{N_T+1} ||\mathbf{y}_{j,\cdot} - \mathcal{D} \circ \mathcal{E}\left(\mathbf{y}_{j,\cdot}\right)||_2^2 \right]_{N_B},$$

$$\mathcal{L}_{dmd} = \left[\frac{1}{N_{lg}} \sum_{p=0}^{N_{lg}-1} \frac{1}{\Delta_p} \sum_{j=N_{st}-p}^{N_w-1} \left|\left| \tilde{\mathbf{y}}_{j,\cdot} - \left(\tilde{\mathbf{Y}}_{N_{st}-p}\right)_{j-(N_{st}-p)+1,\cdot} \right|\right|_2^2 \right]_{N_B},$$

$$\mathcal{L}_{pred} = \left[\frac{1}{N_{lg}} \sum_{p=0}^{N_{lg}-1} \frac{1}{\Delta_p} \sum_{j=N_{st}-p}^{N_w-1} \left|\left| \mathbf{y}_{j,\cdot} - \mathcal{D}\left(\left(\tilde{\mathbf{Y}}_{N_{st}-p}\right)_{j-(N_{st}-p)+1,\cdot} \right) \right|\right|_2^2 \right]_{N_B},$$

with $[\cdot]_{N_B}$ denoting averaging over a given batch, $\Delta_p = N_w - N_{st} + p$, and where we have modified Equation (6) so that

$$\tilde{\mathbf{Y}}_{N_{st}-p} = \mathbf{K}_M \mathbf{L}^{N_{st}-p} \mathbf{\Phi}_-, \ p = 0, \cdots, N_{lg} - 1.$$

The number of lags N_{lg} we introduce can be adjusted so as to reinforce learning dynamics by iterating the eigenvalues which come from EDMD. See [6] for a more complete motivation and discussion of this loss function. We collect the details of our learning method in Algorithm 1, which we call the Deep Learning HDMD (DLHDMD). Note, we perform the update of

Algorithm 1: The DLHDMD Algorithm

Data: Choose parameters N_C, N_B, α, E_{max}, N_{st}, N_{lg}
Data: Choose initial value of N_{ob}
input : N_C trajectories shuffled into *batches* of size N_B
for $l \leftarrow 1$ **to** E_{max} **do**
  for $k \leftarrow 1$ **to** N_B **do**
    $\tilde{y}_{j,k} \leftarrow \mathcal{E}(y_{j,k})$
  Apply the HDMD to $\{\tilde{y}_{j,k}\}_{j,k=1}^{N_w,N_B}$ to generate $\tilde{\mathbf{Y}}_{N_{st}}$;
  $\mathcal{L}_{tot} \leftarrow \mathcal{L}_{recon} + \mathcal{L}_{pred} + \mathcal{L}_{dmd} + \alpha\mathcal{L}_{reg}$;
  Find $\mathcal{E}$ and $\mathcal{D}$ so as to minimize $\mathcal{L}_{tot}$;
  if $0 \equiv l \ mod \ E_{up}$ **then**
    Find minimum of $\mathcal{L}_{dmd}$ for number of observables $N_{ob} - 1, N_{ob}, N_{ob} + 1$ over the validation data;

N_{ob} over the validation data since we typically have $N_C^{(vl)} \ll N_C$, thereby keeping this step relatively economical in terms of computational cost. Also, $\mathcal{L}_{reg}$ is a standard 2-norm regularization of the weights of the auto-encoder.

4.1 Results for DLHDMD

We now show how the DLHDMD performs on several dynamical systems. We take as our training set 10000 randomly chosen initial conditions with their affiliated trajectories, 3000 randomly chosen validation set initial conditions, and 2000 randomly chosen initial conditions for testing purposes. Aside form our results for the KS equation, the training is done over $E_{max} = 100$ epochs using an ADAM optimizer with learning rate $\gamma = 10^{-4}$. The encoder and decoder each consist of five layers consisting of 128 neurons each, and all weights are initially drawn from truncated Gaussian distributions of zero mean and $\sigma = .1$. The batch size $N_B = 256$, and the regularization hyperparamter $\alpha = 10^{-14}$. For the Lorenz-63 and Rossler sytems, we choose the initial number of observables to be $N_{ob} = 10$, and we update every $E_{up} = 5$ epochs. For the KS system, we initially choose $N_{ob} = 5$ and let $E_{up} = 10$. In all cases, we choose $N_{lg} = 1$, which was found to be sufficient for efficient training.

4.1.1 DLHDMD for the Lorenz-63 System

The results of running the DLHDMD for the Lorenz-63 system are found in Figure 5. The maximum positive Lyupanov exponent, say λ_L, for this version of the Lorenz-63 system can be numerically computed, and we find that $\lambda_L \approx .8875$. In this case then, our prediction window is only slightly less than $1/\lambda_L \approx 1.127$, so that we are making predictions up to the point where the strange attractor would tend to induce significant separations in what were initially nearby trajectories. Moreover, as can be seen, the overall reconstruction and forecast, plotted for times t such that $1 \leq t \leq t_{f,w} + 1$, shows excellent agreement with the plot of the Lorenz Butterfly in Figure 3. This degree of accuracy is quantified by the graph of $\mathcal{L}_{pred}$, which shows a relative accuracy of about 1% by the 100^{th} epoch.

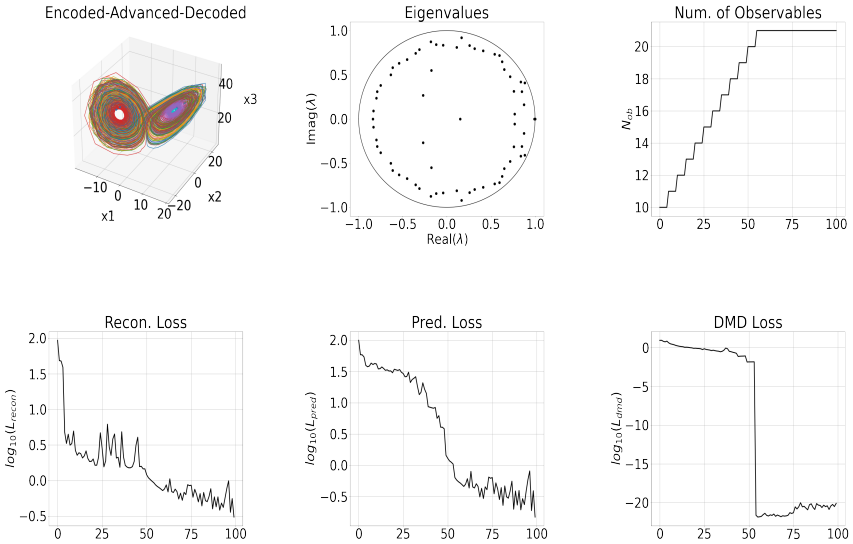


Figure 5: Results of the DLHDMD on the Lorenz-63 system after 100 epochs of training. In the top row, moving from left to right, we see the reconstructed, iterated reconstructed, and forecast data generated by the DLHDMD data, the affiliated spectra from the HDMD, and the plot of N_{ob} over epochs. In the bottom row, moving from left to right, we plot $\mathcal{L}_{recon}$, $\mathcal{L}_{pred}$, and $\mathcal{L}_{dmd}$. Again, the reconstruction is generated for times $1 \leq t \leq t_{f,w}$ and forecasting is done for times $t_{f,w} \leq t \leq t_{f,w} + 1$. Error plots are over validation data.

To achieve this, we see that the DLHDMD progressively raises the value of N_{ob}, thereby adding observables and concomitantly eigenvalues. As seen

in Figure 5, this process continues until about the 50^{th} epoch, at which point $N_{ob} = N_{st}$ and a saturation effect kicks in whereby $\mathcal{L}_{dmd}$ collapses for the given choice of observables. That this is also the point at which we no longer have novel forecasts points to this effect being a kind of overfitting. We note though that if one initially chooses $N_{ob} = N_{st}$, then the training is generally not successful. Thus the model still needs to train to the point at which $N_{ob} = N_{st}$, and it cannot happen too quickly without compromising the success of the training of the machine.

As we increase N_{st} for $N_{lg} = 1$, we see that this same effect occurs when $N_{ob} = N_{st}$. Experiments with $N_{lg} = 5$ showed this collapse in $\mathcal{L}_{dmd}$ continues when $N_{ob} = N_{st}$, though the overall training was stabilized and larger values of N_{st} were able to be used in training. Again, we believe that further exploring the choice of lags through the N_{lg} parameter should help improve this situation, but this will be a subject of future research. Further experiments showed that by setting $E_{up} = 10$, one just delays the epoch at which $N_{ob} = N_{st}$, and until this point is reached, the machine is not able to produce accurate reconstructions, let alone forecasts.

We now look at a typical trajectory both in the original and latent variables to get a better sense of the action of the encoder. As seen in Figure 6, the encoder rescales the data to be more uniform in magnitude across dimensions. However, we also see that the time scale of oscillations are essentially unchanged in the latent relative to the original coordinates. Thus, we see that the HDMD encourages better scaling of the incoming data than necessarily causing any significant changes in the rates of dynamics for the Lorenz-63 system.

4.1.2 DLHDMD for the Rossler System

We now study the Rossler system given by

$$\begin{aligned}\dot{y}_1 =& - y_2 - y_3,\\ \dot{y}_2 =& y_1 + a y_2,\\ \dot{y}_3 =& b + y_3 \left(y_1 - c\right),\end{aligned}$$

where

$$a = .1,\ b = .1,\ c = 14.$$

Aside from the dynamics coalescing onto a strange attractor, the disparity in parameter values gives rise to multiscale phenomena so that there are slow and fast regimes of the dynamics, with the slow portions being approximated by harmonic motion in the (y_1, y_2) plane with fast departures along the y_3 coordinate. This strong disparity in time scales also appears by way of $\lambda_L \approx 1.989$, which is more than double the maximal Lyupanov exponent for the Lorenz-63 system. Thus dynamics separate along the strange attractor twice as fast.

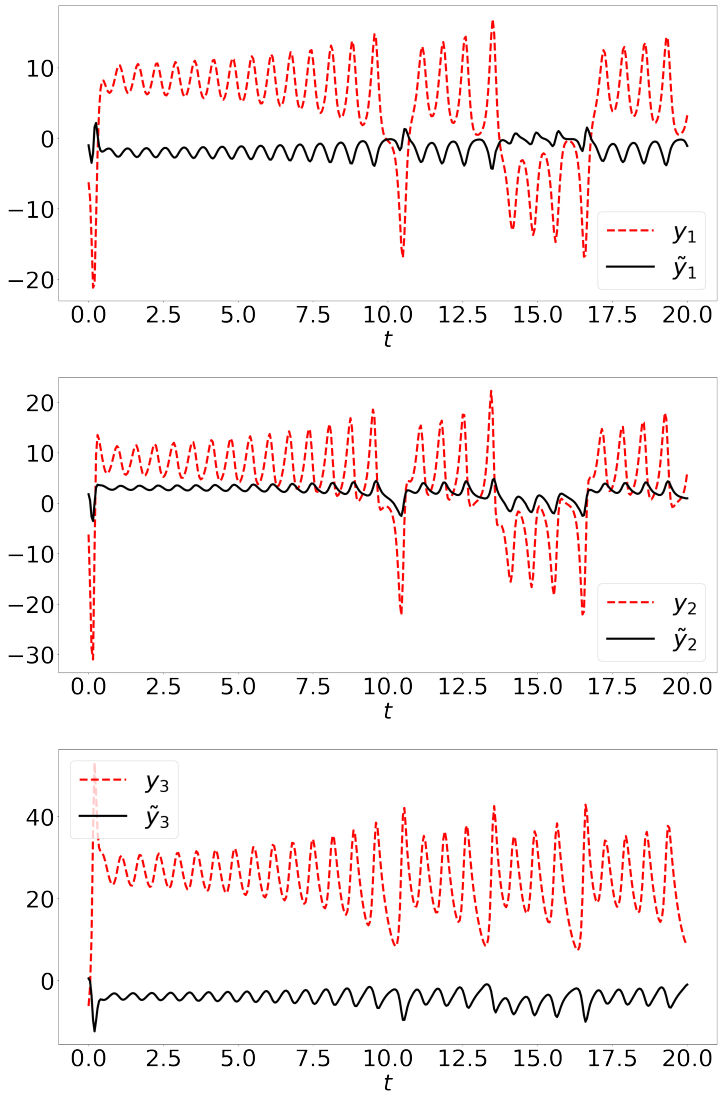


Figure 6: Comparison of original and latent variables for the Lorenz-63 system for a typical test trajectory.

Using then the same parameter choices described above, we get the following results for the training and validation of DLHDMD on the RS; see Figure 7. The performance of DLHDMD is essentially identical to that seen for the Lorenz-63 system. We likewise see the same plummet in the $\mathcal{L}_{dmd}$ term around the 50^{th} epoch mark when $N_{ob} = N_{st}$, though we do see some dynamics in N_{ob} as it seeks to optimize the performance of $\mathcal{L}_{pred}$. Thus we see that our method is able to address slow/fast dynamics with no particular modifications of the algorithm needed. We do note though that a visual inspection of trajectories shows that the error in our model is most apparent

when one is trying to capture the fast transients affiliated with the multiscale dynamics of the Rossler system. Overall though, our iterated reconstruction window is almost twice the length of time over which trajectories separate on the attractor, so the results appear quite good in light of this fact.

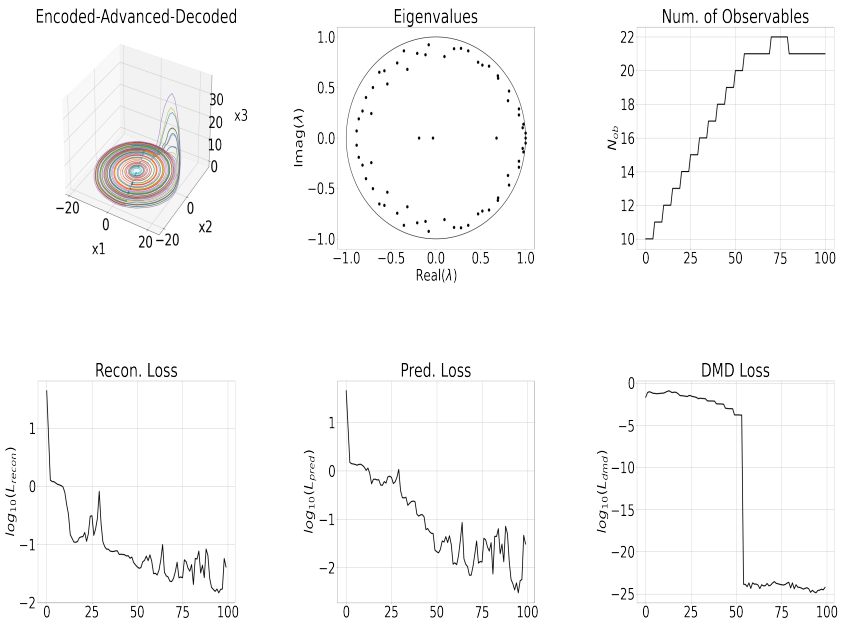


Figure 7: Results of the DLHDMD on the Rossler system after 100 epochs of training. In the top row, moving from left to right, we see the reconstructed, iterated reconstructed, and forecast data generated by the DLHDMD data, the affiliated spectra from the HDMD, and the plot of N_{ob} over epochs. In the bottom row, moving from left to right, we plot $\mathcal{L}_{recon}$, $\mathcal{L}_{pred}$, and $\mathcal{L}_{dmd}$. Again, the reconstruction is generated for times $1 \leq t \leq t_{f,w}$ and forecasting is done for times $t_{f,w} \leq t \leq t_{f,w} + 1$. Error plots are over validation data.

Again, we look at a typical trajectory both in the original and latent variables to get a better sense of the action of the encoder. As seen in Figure 8, the encoder, similar to its effect for the Lorenz-63 system, rescales the data so that it is more uniform across dimensions. However, we also see that fast transients along y_3 are completely removed so that $\tilde{y}_3$ is now a more uniform oscillator. Taking this information together with the Lorenz-63 results, we see the HDMD algorithm guides the learning process to push data to be both more regular in amplitude and the rate of dynamics. Given the linear nature of DMD based algorithms, with their particular focus on

iterating eigenvalues to produce dynamics, makes the latent variable results unsurprising.

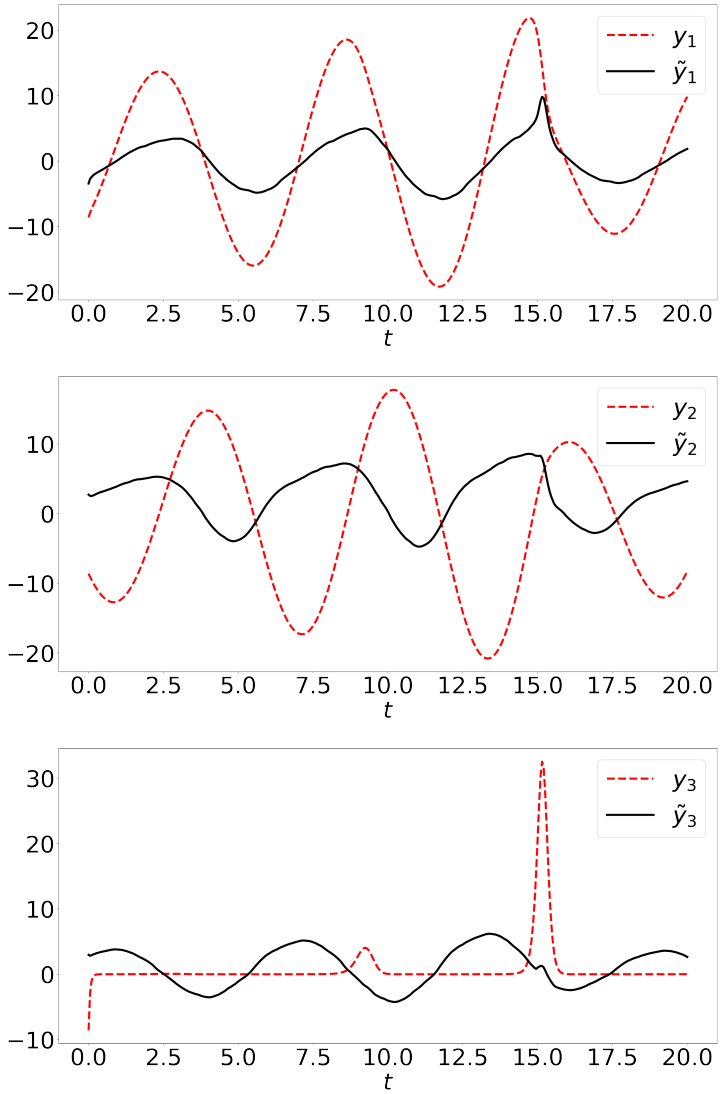


Figure 8: Comparison of original and latent variables for the Rossler system for a typical test trajectory.

4.1.3 DLHDMD for the KS Equation

To see the edges of our method, we now examine spatio-temporal chaos generated by the KS equation with periodic-boundary conditions in the form

$$u_t + u_{xx} + u_{xxxx} + uu_x = 0,\ u(x + 2L, t) = u(x, t).$$

Note, given the vast size of the literature around the KS equation, we refer the reader to [23] for an extensive bibliography with regards to details and pertinent proofs of facts used in this section. Introducing the rescalings

$$\tilde{t} = \frac{t}{T},\ \tilde{x} = \frac{\pi}{L}x,\ u = A\tilde{u},$$

and taking the balances

$$A = \frac{L}{\pi T},\ T = \left(\frac{L}{\pi}\right)^2,$$

we get the equivalent KS equation (dropping tildes for ease of reading)

$$u_t + u_{xx} + \nu u_{xxxx} + uu_x = 0,\ \nu = \left(\frac{\pi}{L}\right)^2.$$

Looking at the linearized dispersion relationship $\omega(k) = k^2 - \nu k^4$, we see that the ν parameter acts as a viscous damping term. Thus, as the system size L is increased, the effective viscosity is decreased, thereby allowing for more complex dynamics to emerge. As is now well known, for L sufficiently large, a fractional-finite- dimensional-strange attractor forms which both produces intricate spatio-temporal dynamics while also allowing for a far simpler representation of said dynamics. It is has been shown in many different works (see for example [24]) that $L = 11$ generates a strange attractor with dimension between eight and nine, and that this is about the smallest value of L which is guaranteed to generate chaotic dynamics. We therefore set $L = 11$ throughout the remainder of this section.

To study the DLDHMD on the KS equation, we use KS data numerically generated by a pseudo-spectral in space and fourth-order exponential-differencing Runge-Kutta in time method [25] of lines approach. For the pseudo-spectral method, K=128 total modes are used giving an effective spatial mesh width of $2L/K = .172$, while the time step for the Runge-Kutta scheme is set to $\delta t = .25$. These particular choices were made with regards to practical memory and simulation time length constraints. After a burn in time of $t_b = (L/\pi)^4 = 150.3$, which is the time scale affiliated with the fourth-order spatial derivative for the chosen value of L, 15000 trajectories of total simulation time length $t_f = (L/\pi)^4$ were used with gaps of L/π in between to allow for nonlinear effects to make each sample significantly different from its neighbors. Each of the 15000 space/time trajectories was then separated via a POD into space and time modes; see [26]. Taking $N_s = 12$ modes captured between 97.8% and 99.4% of the total energy. The choice of the total time scale t_f also ensured that the ratio of the largest and smallest singular values affiliated with the POD was between $10^{-1.1}$ and $10^{-1.9}$ so that the relative importance of each of the modes was roughly the same across all samples. We take this as an indication that each

12-dimensional affiliated time series is accurately tracing dynamics along a common finite-dimensional strange attractor as expected in the KS equation. Using the methods of [27], we can find across batches that typically the largest positive Lyupanov exponent $\lambda_L \approx .3930$, so that $1/\lambda_L \approx 2.545$ is the time after which we anticipate the strange attractor starting to fully pull trajectories apart.

With regards to the details of the DLHDMD, we again use 10000 samples for training, 3000 for validation, and 2000 for testing. The best results with regards to window size were found when we initially set $N_{ob} = 5$ and $E_{up} = 10$. The iterated reconstruction/forecast horizon determined by the choice of N_{st} was chosen so that $N_{st} = (L/\pi)/\delta t \approx 14$, corresponding to the time scale over which nonlinear advection acts. Thus, reconstruction is done on each sample for values of t such that $L/\pi \leq t \leq t_{f,w}$, and iterated reconstruction/forecasting is done for t such that $t_{f,w} \leq t \leq t_{f,w} + L/\pi$. Note, for our initial choice of N_w, we have that initially $t_{f,w} = (L\pi)^4 - 1.25$. The results of DLHDMD training on the $N_s = 12$ dimensional POD reduction of the KS dynamics is shown in Figures 9 and 10. Likewise, our prediction window is longer than the timescale set by λ_L, so we argue our forecasts are over time scales for which chaotic effects are significant.

We see that while the reconstruction and predictions appear accurate; see in particular the comparisons in Figure 10. The collapse of the DMD approximation seen in the previous examples above is now absent, though we see that N_{ob} has just reached N_{st} in our simulations. Thus, by using a window update that is half the rate used in the prior systems, we avoid the affiliated overfitting seen in the prior cases, though we should anticipate that it would probably occur with a few more training epochs.

5 Mutual Information for Characterizing Embeddings

Given the success of the DLHDMD in reconstructing and forecasting dynamics along a strange attractor, especially when compared to the relative failure of trying to do the same using just the HDMD alone, it is of further interest to try to assess exactly what role the auto-encoder plays in improving the outcome of the HDMD. While we can certainly point to the performance of the components of the loss function $\mathcal{L}_{tot}$ to explain the impact of the encoder, this does not provide us with any more explanatory power. In [6], it was empirically shown that the role of the encoder was to generally transform time series to nearly monochromatic periodic signals, which is to say, the effect of encoding was to generate far more localized Fourier spectral representations of the original time series. This does not turn out to be the case though for the DLHDMD. Instead, inspired both by the evolving understanding of how mutual information better explains

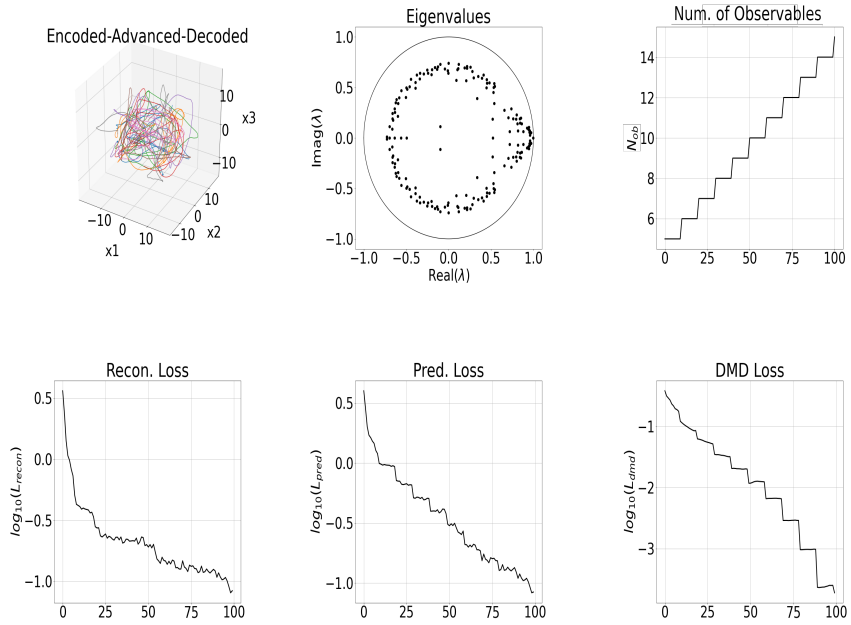


Figure 9: Results of the DLHDMD on the KS system after 100 epochs of training. In the top row, moving from left to right, we see the reconstructed, iterated reconstructed, and forecast data generated by the DLHDMD data, the affiliated spectra from the HDMD, and the plot of N_{ob} over epochs. In the bottom row, moving from left to right, we plot $\mathcal{L}_{recon}$, $\mathcal{L}_{pred}$, and $\mathcal{L}_{dmd}$. Again, the reconstruction is generated for times $L/\pi \leq t \leq t_{f,w}$ and forecasting is done for times $t_{f,w} \leq t \leq t_{f,w} + L/\pi$. $t_{f,w}$ is initially $(L\pi)^4 - 1.25$. Error plots are over validation data.

results in dynamical systems [1, 28] and machine learning [16, 17], we assess the impact of the encoder on the DLHDMD by tracking how the information across dimensions and time lags changes in the original and latent variables.

For two random variables $\mathbf{X}$ and $\mathbf{Y}$ with joint density $p(\mathbf{X}, \mathbf{Y})$, the mutual information (MI) between them $I(\mathbf{X}, \mathbf{Y})$ is defined to be

$$I(\mathbf{X}, \mathbf{Y}) = \int p(\mathbf{x}, \mathbf{y}) \log \left(\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right) d\mathbf{x}d\mathbf{y},$$

where $p(\mathbf{X})$ and $p(\mathbf{Y})$ are the affiliated marginals. One can readily show that $I(\mathbf{X}, \mathbf{Y}) \geq 0$ and $I(\mathbf{X}, \mathbf{Y}) = 0$ if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent. Thus information gives us a stronger metric of statistical coupling between random variables than more traditional tools in time series analysis such

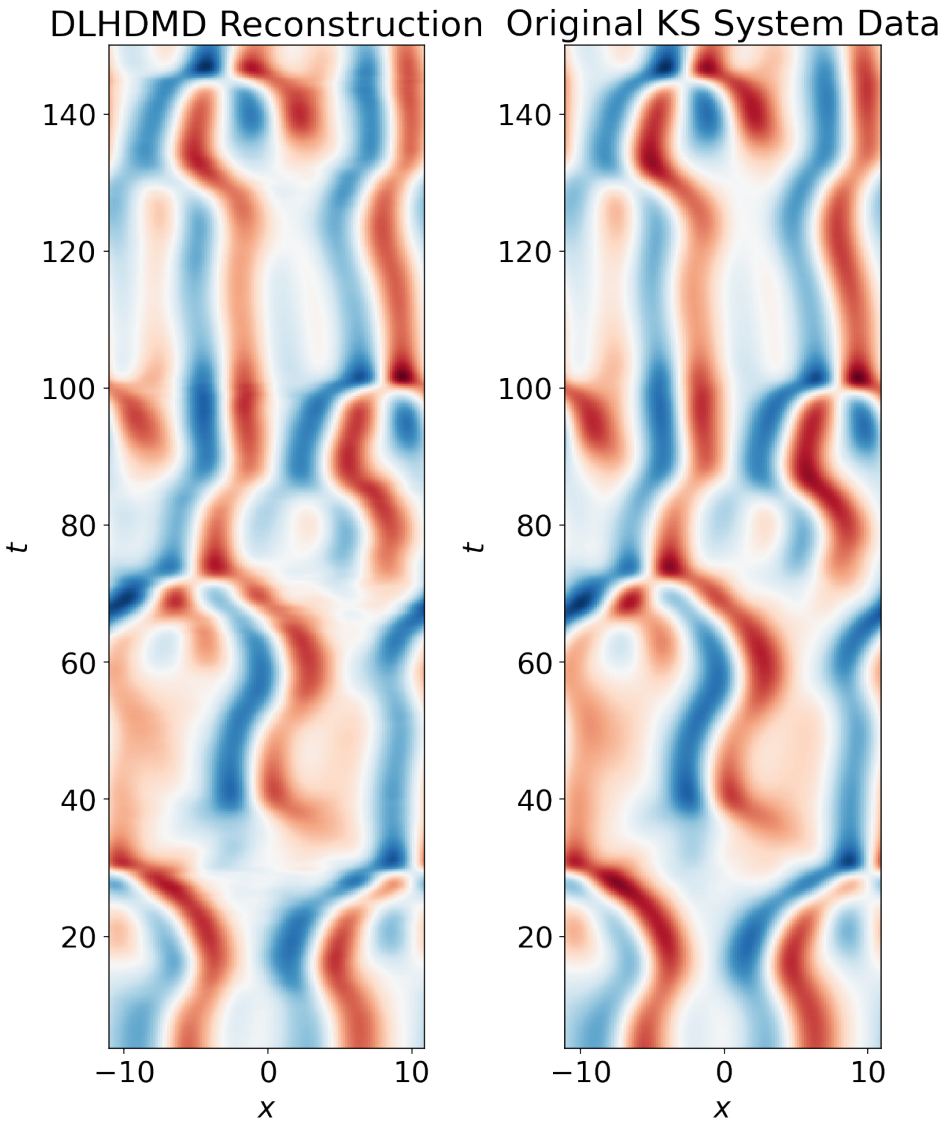


Figure 10: Comparison of DLHDMD results and original KS data.

as correlation measurements. We also should note here that $I(\mathbf{X}, \mathbf{Y}) = I(\mathbf{Y}, \mathbf{X})$, which is to say it is symmetric. We also note that MI is invariant under the action of diffeomorphisms of the variables. Thus we cannot expect to get much use from computing the multidimensional MI of the original and latent variables, thereby allowing for meaningful differences to appear between original and latent variable computations.

Instead, using the $N_C = 2000$ trajectories in the test data, we define the m-step averaged lagged self-information (ALSI) between the n^{th} and v^{th} dimensions $I_{nv}(m)$ to be

$$I_{nv}(m) = \frac{1}{N_C} \sum_{k=1}^{N_C} I\left(y_{n,\cdot,k}, y_{v,\cdot+m,k}\right).$$

We refer to the parameter m as a *lag*. In words then, after averaging over the ensemble of initial conditions in the test data, we compute the degree to which the signal becomes statistically independent from itself across all of the dimensions along which the dynamics evolve. We emphasize that due to the strong nonlinearities in our dynamics, we compute the lagged information as opposed to the more traditional auto-correlation so as to get a more accurate understanding of the degree of self-dependence across dimension in our dynamics. Further, by measuring the lagged MI across isolated dimensions, we break the invariance of MI with respect to diffeomorphisms.

5.1 MI for the Lorenz-63 System

The results of computing the ALSI for the Lorenz-63 system are plotted in Figure 11. As can be seen, the impact of the encoder is to either weakly attenuate the dependency between dimensions; see I_{11} and I_{22}. For I_{33}, the encoder leaves the ALSI essentially unchanged. Finaly, we also see significant phase shifts in the lag count; see I_{13} and I_{23}. In these phase shifts, we see that the shift is always left towards shorter lags, so that the dependence in the latent variables decays more rapidly than in the original variables. In this sense then, the overall tendency of the encoder is to either reduce MI or cause time series to become more independent more rapidly. Otherwise though, the timescales of oscillation in the latent variables are essentially identical to those seen in the latent variables, which is confirmed by comparisons of the original and latent variable dynamics presented in Figure 6. In terms of the DLHDMD, we might then say that the encoder assists the HDMD by generally making the rows of the affiliated Hankel matrices more independent, especially over longer time scales, and therefore more meaningful with regards to their generating more accurate approximations of the underlying Koopman operator.

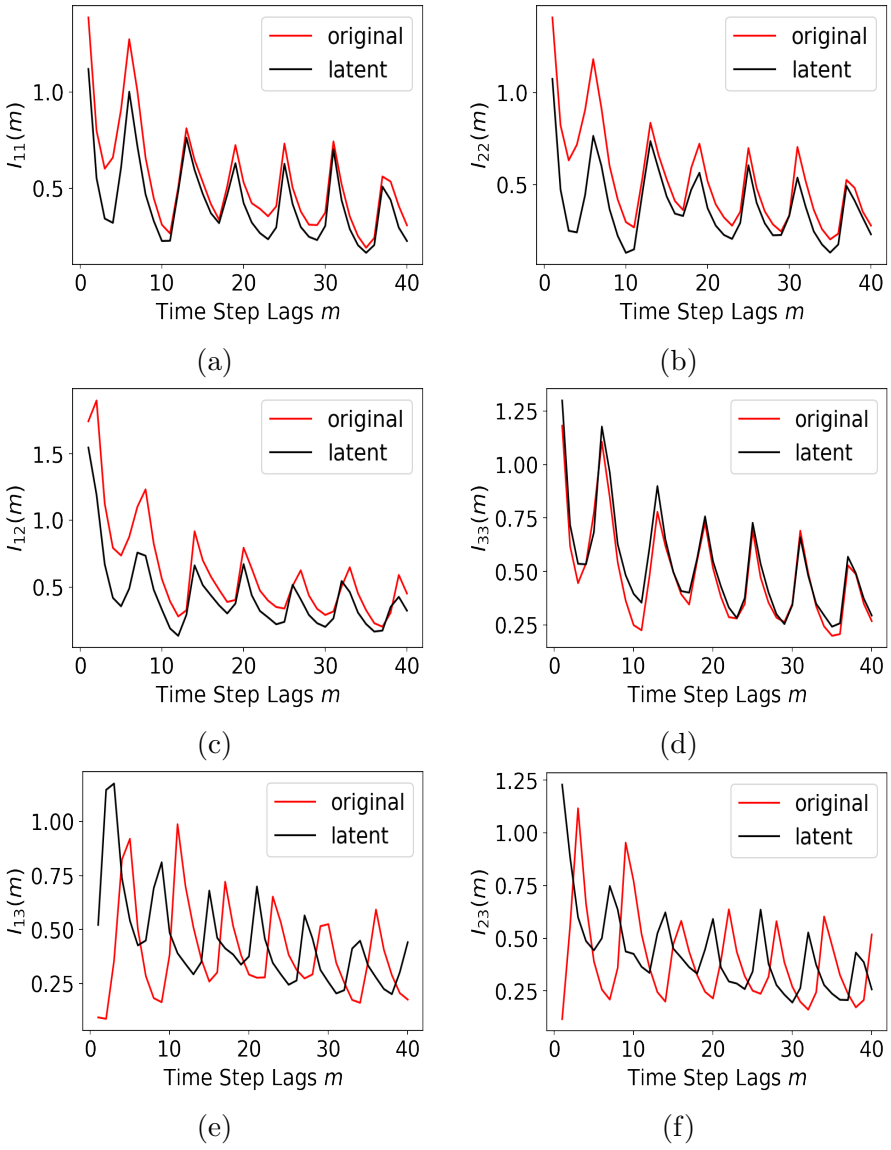


Figure 11: For the Lorenz-63 system, plots of the ALSI $I_{nv}(m)$ for $(n, v) = (1, 1)$ (a), $(n, v) = (2, 2)$ (b), $(n, v) = (1, 2)$ (c), $(n, v) = (3, 3)$ (d), $(n, v) = (1, 3)$ (e), and $(n, v) = (2, 3)$ (f) for both the original and latent coordinates. As can be seen, the encoder tends to reduce the ALSI along each physical dimension aside from those involving the third physical dimension, for which the ALSI is enhanced for shorter lags and decreased for longer ones.

5.2 MI for the Rossler System

When we examine the evolution over lags of the ALSI, we see in Figure 12 that the encoder is causing large and significant changes to the dynamics. In

particular, as we might expect from looking at the comparisons in Figure **??**, when we look at the plots of I_{12} and I_{23}, we see that the sharp transients in the ALSI for the original coordinates is removed and the overall ALSI is relatively flattened in the latent coordinates. This would seem to indicate that the slow/fast dichotomy in the Rossler dynamics is removed and so made more uniform. Also of note though is I_{13} which shows that the dependency between the $\tilde{y}_1$ and $\tilde{y}_3$ axes is enhanced relative to the coupling between y_1 and y_3 and that said dependency increases with lags. This reflects the more uniform coupling across dimensions in the latent variables which was seen in Figure 8.

6 Conclusion and Discussion

In this work, we have developed a machine learning enhanced version of the HDMD which we call the DLHDMD. We have shown that its performance is significantly better than just the HDMD method alone, and when comparing against existing results in [6] we see radical improvement over the DLDMD method for the Lorenz-63 system. Likewise, we find that our method is successful across several challenging chaotic dynamical systems varying in dynamical features and size. Thus, we have a parallel approach of similar accuracy fitting within the larger framework of Koopman operator based methods. Moreover, we have a method which computes Koopman modes globally and naturally localizes spectra around the complex unit circle without further control of the method. Finally, our analysis of the relative information dynamics across physical dimensions in the original and latent variables provides us a means of understanding the impact of the encoder network on the dynamics in line with modern thinking in machine learning as well as better pointing towards an understanding that the HDMD is enhanced by decreasing the relative statistical dependence across physical dimensions.

As explained in detail in the Introduction, there are of course a number of questions that remain to be addressed, and they will certainly be the subject of future research.

References

[1] A. M. Fraser and H.L. Swinney. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, 33:1134–1140, 1986.

[2] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *J. Stat. Phys.*, 65:579–616, 1991.

[3] W. Gilpin. Deep reconstruction of strange attractors from time series. NeurIPS, 2020.

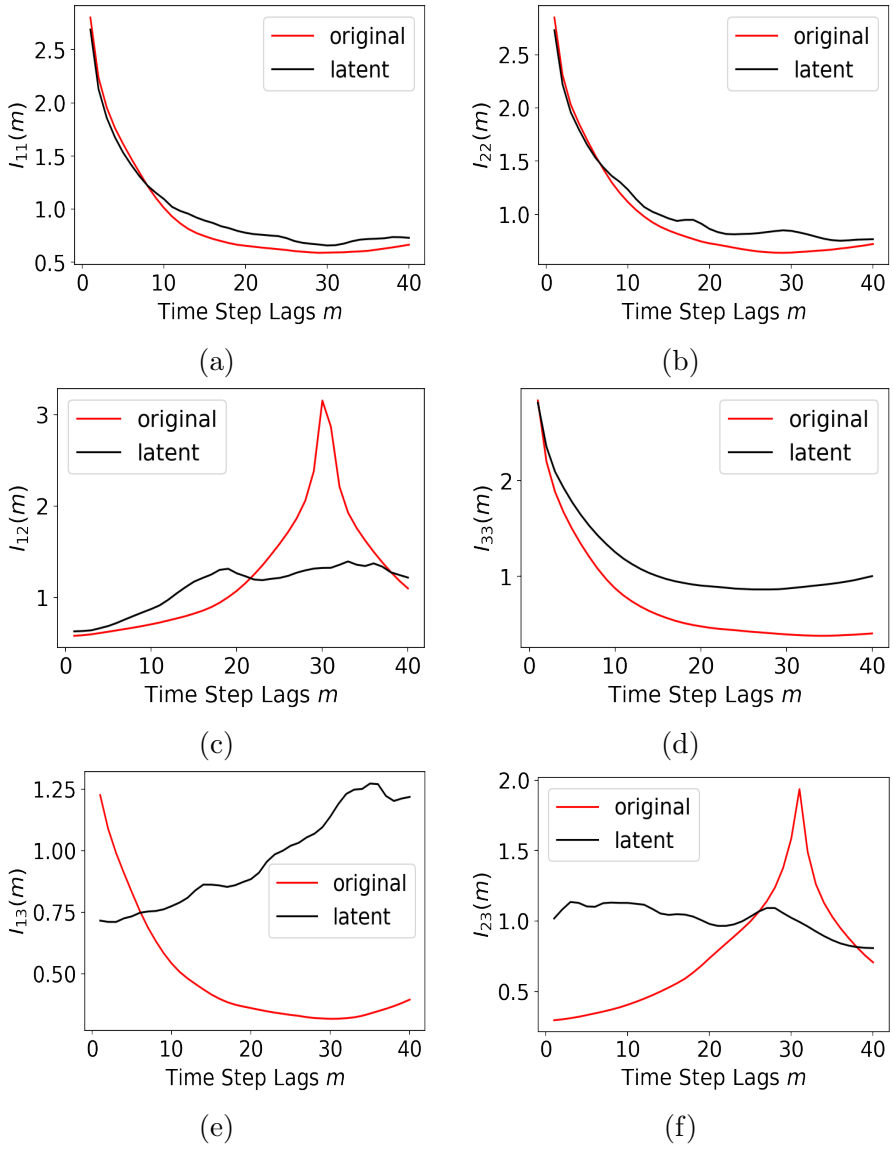


Figure 12: For the Rossler system, plots of the ALSI $I_{nv}(m)$ for $(n, v) = (1, 1)$ (a), $(n, v) = (2, 2)$ (b), $(n, v) = (1, 2)$ (c), $(n, v) = (3, 3)$ (d), $(n, v) = (1, 3)$ (e), and $(n, v) = (2, 3)$ (f) for both the original and latent coordinates.

[4] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Comm.*, 9:4950, 2018.

[5] Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119

of *Proceedings of Machine Learning Research*, pages 475–485. PMLR, 13–18 Jul 2020.

[6] D. J. Alford-Lago, C. W. Curtis, A. T. Ihler, and O. Issan. Deep learning enhanced dynamic mode decomposition. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32:033116, 2022.

[7] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[8] K. Kadierdan, J.N. Kutz, and S.L. Brunton. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. Roc. Soc. A*, 476:20200279, 2020.

[9] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019.

[10] Zongyi Li, Hongkai Zheng, Nikola Borislavov Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Andrew Stuart, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations, 2022.

[11] S.L. Brunton, B.W. Brunton, J.L. Proctor, , E. Kaiser, and J.N. Kutz. Chaos as an intermittently forced linear system. *Nature Comm.*, 8, 2017.

[12] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[13] E. Bollt. On explaining the surprising success of resevoir computing forecaster of chaos? The universal machine learningn dynamical system with contrast to var and dmd. *Chaos*, 31:013108, 2021.

[14] D.J. Gauthier, E. Bollt, A. Griffith, and W.A.S. Barbosa. Next generation resevoir computing. *Nat. Comm.*, 12:55674, 2021.

[15] H. Arbabi and I. Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. *SIAM Appl. Dyn. Sys.*, 16:2096–2126, 2017.

[16] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[17] O. Calin. *Deep Learning Architectures: A Mathematical Approach.* Springer, Cham, 2020.

[18] G.F. Fasshauer. *Meshfree Appoximation Methods with Matlab.* World Scientific, Hackensack, NJ, 2007.

[19] M.O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *J. Nonlin. Sci.*, 25:1307–1346, 2015.

[20] M.O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based method for data driven Koopman spectral analysis. *J. Comp. Dyn.*, 2:247–265, 2015.

[21] F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, pages 366–381. 1981.

[22] D.S. Broomhead and G.P. King. Extracting qualitative dynamics from experimental data. *Physica D*, 20:217–236, 1986.

[23] J.C. Robinson. *Infinite Dimensional Dynamical Systems.* Cambridge University Press, Cambridge, UK, 2001.

[24] Nazmi Burak Budanur, Predrag Cvitanović, Ruslan L. Davidchack, and Evangelos Siminos. Reduction of so(2) symmetry for spatially extended dynamical systems. *Phys. Rev. Lett.*, 114:084102, Feb 2015.

[25] AK Kassam and L.N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM J. Sci. Comp.*, 26:1214–1233, 2005.

[26] G. Berkooz, P. Holmes, J.L. Lumley, and C.W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems, and Symmetry.* Cambridge University Press, Cambridge, UK, 2012.

[27] H.D.I. Abarbanel, R. Brown, and M.B. Kennel. Local Lyupanov exponents computed from observed data. *J. Nonlinear Sci.*, 2:343–365, 1992.

[28] E.M. Bollt and N. Santitissadeekorn. *Applied and Computational Measurable Dynamics.* SIAM, Philadelphia, PA, 2013.