

The structure of reconstructed flows in latent spaces

Gonzalo Uribarri¹ and Gabriel B. Mindlin^{1,*}

¹IFIBA, CONICET and Departamento de Física, FCEyN, UBA, Buenos Aires, Argentina

*corresponding author at: IFIBA, CONICET and Departamento de Física, FCEyN, UBA, Ciudad Universitaria, pabellon 1, cp 1428, Buenos Aires, Argentina.
e-mail: gabo@df.uba.ar

Abstract

Reconstructing the flow of a dynamical system from experimental data has been a key tool in the study of nonlinear problems. It allows to discover the equations ruling the dynamics of a system, as well as to quantify its complexity. In this work we study the topology of the flow reconstructed by autoencoders, a dimensionality reduction method based on deep neural networks that has recently proved to be a very powerful tool for this task. We show that, although in many cases proper embeddings can be obtained with this method, it is not always the case that the topological structure of the flow is preserved.

Lead Paragraph

One of the main objectives in science and engineering is to propose interpretable mathematical models capable of explaining sequences of experimental observations. These models are usually expressed as a set of differential equations. The explosive increase of computing power and data availability in recent years has boosted the development of machine learning algorithms capable of find these equations directly from the data (1-7). This is especially relevant when we do not know the elemental mechanisms that govern the evolution of the system, or when the nature of the observations does not allow a direct analytical approach. Recent work shows that this data-driven methodology for discovering governing equations can greatly benefit from deep neural networks (8-11). These powerful models can be used to transform the experimental data into a new set of coordinates in which the dynamics can be easily expressed. This process is known as embedding.

In this paper we study reconstructed flows using autoencoders, a neural network architecture widely used for dimensionality reduction in machine learning. For most cases, this model is capable of learning proper embeddings of chaotic data. But, remarkably, there are cases where this model does not learn a topologically correct representation of the data, even when its reconstruction error is low. Autoencoders have the potential to drastically improve equation discovery methods, but a warning should be

raised since it is not possible to find a correct model if topological invariants computed from the data are not preserved.

Introduction

In all areas of physics, we need to build and validate models involving more variables than those available for measurement. Some forty years ago, Floris Takens (12) presented a theorem which described the conditions under which a flow can be reconstructed from a temporal sequence of observations of a system's state. Moreover, he presented two ways to create a multi-valued sequence of points which could be mapped into the original flow by means of a smooth and invertible change of coordinates. That reconstruction was used to compute the system's effective dimension, to reconstruct the equations driving its dynamics, and it was even successful in preserving, for a chaotic flow, the topological organization of the unstable orbits coexisting with the attractor (13). This is particularly useful, since a model proposed to explain an experiment can be discarded if it is incapable of reproducing the flow's topology (14).

Due to the increase of data availability and computing power, data-driven algorithms for discovering interpretable dynamical models have become highly relevant in the last decade. Model discovery algorithms can be very useful in systems where the governing equations are unknown, or only partially known, but rich time series data is available. This is the case for many problems in biological and social sciences, or many control and stabilization tasks in fluid dynamics (15,16). Many research efforts have been developed to investigate effective algorithms capable of finding parsimonious, nonlinear governing equations from temporal data series (1-7). Before applying these algorithms, an embedding procedure is usually necessary to transform the high-dimensional input data to the low-dimensional coordinate space of the equation system. The standard procedure is to use some kind of mode decomposition procedure (3,17,18).

Recently, the use of deep autoencoders for discovering nonlinear coordinate transformation has proved to be a very successful approach to make the embedding (8-11). Such novel scheme takes advantage of the well-known power of neural networks to efficiently discover complex patterns in large amounts of data (19,20). This method has the potential to provide qualitative improvements in our understanding of a variety of complex systems, whose equations are difficult to derive from first principles. Yet, a key question in this program is whether these reconstructed variables will reflect the topological structure of the original flow, as the reconstructions proposed by Takens do.

Let us assume we are interested in unveiling the structure of a flow, whose measurement gives rise to a time series data $\{x_1, x_2, \dots, x_N\}$. The underlying hypothesis is that $N_1 \ll N$ elements of the time series $\{x_k, x_{k+1}, \dots, x_{N_1+k-1}\}$ contain enough information to define its future, i.e., the next segment $\{x_{k+1}, x_{k+2}, \dots, x_{N_1+k}\}$. Determinism will require that there is a unique value x_{N_1+k} for any existing segment. In this way, we can represent the

points $\{x_i\}$ in the time series by points in R^{N_1} . Hence, the time series will be restricted to a submanifold of R^{N_1} : an embedded manifold. This embedding is neither optimal nor unique. In fact, Taken's theorem provides a subsampling that guarantees global metric embedding between original and reconstructed state spaces, and it follows from it that *if the manifold holding the flow is of dimension d , under general conditions a subsampled set of $2d + 1$ points of the N_1 segments suffices to generate an embedding*. This is how dynamics typically deals with the reconstruction of flows from data.

The autoencoder embedding method reconstructs the flow from the activity of the artificial neurons in the middle layer of a deep autoencoder, known as latent space. The term autoencoder refers to a feedforward neural network that learns to copy its input to its output (20-23). Usually, it consists of a set of layers of nonlinear units, where the number of units in each layer is:

$$N_1, N_2, \dots, N_k, \dots, N_2, N_1 \quad \text{with} \quad N_1 > N_2 > \dots > N_k.$$

When the number of hidden layers is greater than one, these neural networks are usually referred to as stacked or deep autoencoders. What makes the network interesting is that N_k is chosen as small as possible, i.e., as *the minimum number of units that prevents the loss of information by the network*, leading to a compression of the information of the original data. We can train the network with subsets of the time series,

$$\{x_1, x_2, \dots, x_{N_1}\}, \{x_2, x_3, \dots, x_{N_1+1}\}, \dots, \{x_{N-N_1}, x_{N-N_1+1}, \dots, x_N\}.$$

If N_1 is large enough so that determinism is guaranteed, the next question is how small can N_k be, so that the compression does not affect the network's capacity to recover the input information in the last layer. Once that value for N_k is obtained, the middle layer, with N_k units, can be used to define a compressed multi-valued environment for each input element:

$$(x_1^i, x_2^i, \dots, x_{N_1}^i) \rightarrow Y^i = (y_1^i, y_2^i, \dots, y_{N_k}^i), \quad i = 1, \dots, N - N_1.$$

The question that we address now is whether this multi-valued, compressed set of vectors constitutes an embedding of the scalar time series. Specifically, if the scalar values of the time series corresponding to flow measurements, processed by an autoencoder, give rise to a multi-valued environment *preserving the topology of the flow*. Notice that the input needs not to be a scalar time series. It could be a tensor, where at each time step a frame is coded. In this way, an embedding of the dynamics of phenomena recorded in a movie could be automatically reconstructed, without the need of computing empirical modes, and projecting the spatio-temporal data onto them.

In order to address this question, we will study strange attractors: non-trivial invariant sets whose topology can be very precisely quantified through the study of the unstable periodic orbits coexisting with it (24). Strange attractors can be classified by the

topological organization of those periodic orbits. For attractors in three dimensions, this organization can be characterized by how the orbits are knotted, and how they link around each other (13,14). The later topological feature can be described by an index called the linking number. Given two orbits in a three-dimensional space, this index can be computed algorithmically from the crossings observed in a two-dimensional projection of the orbits (14). At each crossing point, tangent vectors are drawn to the upper and lower curve segments in the direction of the flow. The crossing is labelled + (-) if the tangent vectors form a right- (left-) handed coordinate system in the projection plane (see Figure 1). The linking number of these two orbits is the integer resulting from the sum over all crossings, where each crossing contributes $+1/2$ or $-1/2$, depending on its sign.

Remarkably, flows reconstructed through time delay embeddings as proposed by Takens preserve the topological structure of time series segments which are good approximations of the unstable periodic orbits (13,14). Analogously, we used the variables in the latent space of an autoencoder to construct a flow and test whether this methodology also preserves the topological organization of the unstable periodic orbits.

Methods

To generate our dataset, we integrate a Rössler system of equations,

$$\begin{aligned}\frac{dx}{dt} &= -y - z \\ \frac{dy}{dt} &= x + ay \\ \frac{dz}{dt} &= b + z(x - c)\end{aligned}$$

with $a = b = 0.1$ and $c = 14$, generating a time series data of 15000 scalar values corresponding to consecutive values of x (see Figure 2a). We partitioned the time series in segments of $N_1 = 256$ points, corresponding to 10.26 integration units, as shown in Figure 2b: $\{x_1, x_2, \dots, x_{N_1}\}$, $\{x_2, x_3, \dots, x_{N_1+1}\}$, ... $\{x_{N-N_1}, x_{N-N_1+1}, \dots, x_N\}$. This time window was chosen in a way consistent with Taken's criterium. A typical time delay would be about one third of the shortest characteristic recurrence, and the system was not expected to need more than five dimensions for its reconstruction. That led us to consider a temporal window of at least $4/3$ of the duration of the time series segment approximating the period one orbit. This process resulted in 374488 elements that we used to train and test an autoencoder. The first seventy percent of the data was used to train the network, and the last thirty percent was used to test it.

The network was implemented on Keras version 2.3.1, backened by Tensorflow 1.15.2. Each neuron was modelled as a ReLU unit, except for the neurons in the middle layer which were modelled as linear units (25). The minimization of the Mean Square Error (MSE) between the entry and exit layers was used to train the network, through an Adam

algorithm with learning rate $lr = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ (26). The batch size was 32 samples and a Glorot uniform initializer was used for weight initialization (27). Autoencoders were trained for 91 epochs, this stopping criterion was chosen to guarantee that the average percentage change of the test MSE was lower than 1%. We did not perform a systematic search in the hyperparameter space to optimize the MSE on the test set.

The values of the variables representing the three neurons of the middle layer were used as a multi-valued environment: our prospective new embedding. In this particular case, we already know that the minimum number of dimensions in which we can express the underlying dynamical system is 3. In a general situation, one can choose the dimension of the latent space by studying the reconstruction MSE. For example, in our case, evaluating networks with 2,3,4 and 5 shows that a qualitative improvement occurs when we use three units or more: the MSE for two units is one order of magnitude larger than the MSE obtained with three units. In other words, two units in the middle layer do not allow a proper reconstruction of the flow, since the loss of information is too large (see Figure 2d).

To numerically compute the Linking Number between two three-dimensional oriented curves, as explained in the previous section (See Figure 1), we programmed a function in Python language, the source code is available on a public repository (28). To estimate the Lyapunov exponents we used the numerical methodology proposed by Eckmann and Ruelle (29,30). We implemented the algorithm in python, based on code taken from the Nonlinear Measures for Dynamical Systems library (31).

Results

Out of the numerical simulations of our dynamical system (see Figure 3a), we singled out three segments which were good approximations of unstable periodic orbits coexisting with the chaotic solution obtained in our numerical experiment (see Figure 3b). We defined a Poincare section $\{y = 0\}$, and first, we looked for a segment of the original flow where the distance between two consecutive intersections of the flow with the section was a minimum (period 1, solution P1). We repeated the procedure for an orbit crossing the section twice (period 2, solution P2), and three times (a period 3, solution P3, which is one of the two solutions of period three that can be found in the Rössler system). We numerically computed the Linking Numbers between these approximations of periodic orbits in the Rössler system, obtaining: $Linking(P1, P2) = -1$, $Linking(P1, P3) = -1$, $Linking(P2, P3) = -2$.

We compared this organization of the segments in the original phase space, with the organization of the segments in the latent space (i.e., the multi-valued environment where the coordinates are the activities of the neurons in the middle layer of the autoencoder). In Figure 3d, we display this multi-valued environment for the whole time series for one of the sixty trained networks, and in Figure 3e, we display the reconstruction for the three

segments approximating periodic solutions in the original flow. In this example the autoencoder training leads to a correct embedding: the periodic orbits in the latent space have a topological organization identical to that of the original orbits (see Figure 3e).

It is interesting to explore how the neural network achieves this embedding of the time series. We inspected the topological structure of the orbits, reconstructed from the variables of the latent space at different epochs during the training procedure, for sixty different autoencoders. In the first epoch, the sixty autoencoders gave rise to wrong topologies (See Figure 4a). At successive epochs, the mean square errors (MSE), which indicate the difference between the input and output layers, typically decrease, and the percentage of networks leading to a correct topology increases. After ninety epochs, 53 models lead to the right topological organization between the reconstructed orbits, whereas 7 do not. To ensure we were not cutting the training too soon, we continued training models with incorrect topology for 90 more epochs. None of them changed its topology during this process. In Figure 4b we show the evolution of the MSE during the training procedure for the 60 models segregated by topological structure. Notice that the MSE typically reaches a plateau after ninety epochs. In Figure 4c we show the linking numbers between the three orbits for different epochs. In this particular numerical experiment, the autoencoder leads to a proper reconstruction of the topology after twenty-nine epochs.

Remarkably the majority of the autoencoders provide the right topological organization between the reconstructed orbits. But it is also important to notice that more than 10% of them do not, even after their MSE has reached a plateau, a benchmark commonly used to end the training procedure. Moreover, models that do not constitute a proper embedding can present a much lower MSE than that of others that lead to a correct one (see error bars in Figure 4b).

It is interesting to analyse what happens in the reconstructions that lead to an incorrect topology. To this end, we explored whether the vectors describing the flow at nearby points in the reconstructed space were mostly parallel (32). We computed the cosine of the angle formed by directions of the flow in neighbouring trajectories for 25000 points in the test set. For the majority of the points, this value is very close to 1, indicating almost parallel directions (i.e., nearby points share similar futures). But, if we look at the 50 points with the lowest values in each autoencoder, there is a significant difference between those with correct topology, 0.93 (0.88 - 0.96), and those with an incorrect one 0.79 (0.46-0.87). Values are reported as median (interquartile range). This result suggests that there are regions in the latent space where autoencoders leading to wrong topologies present self-intersections in the reconstructed trajectories, since points in a small box of the phase space display qualitatively different futures (32). This does not happen in those that lead to a correct topology.

Strange attractors can also be characterized by the Lyapunov exponents (30), which describe the sensitivity of the flow to different initial conditions and allow to reconstruct

metric properties as the flow's dimension. In Table 1 we present the values of the first Lyapunov exponent for the Rössler attractor in the original space and in the latent space. Unlike linking numbers, the values of the Lyapunov exponents change for different system's parameters (a,b and c). We trained 20 autoencoders with data coming from systems with two other sets of parameters, and the results are also shown in Table 1. These show that the reconstructed attractors with correct topology present values close to the one computed for the original attractor, while those with incorrect topology present significantly higher values.

A natural question is whether these results hold for time series with noise. To explore this issue, we trained 20 autoencoders using the same methodology detailed in methods, but adding noise to the input time series x of the Rössler equations. The noise was Gaussian white noise, with zero mean, and a standard deviation equivalent to 5% of the standard deviation of x . When analysing the topology of the reconstructed attractors, we found that 17 autoencoders (85%) gave rise to the right topology, a ratio very similar to the one obtained with non-noisy time series data. Although this subject requires further investigation, the autoencoder embedding methodology seems to be robust under noise.

Discussion

Takens' embedding theorem has been a breakthrough in our understanding of nonlinear systems. Reconstructed attractors through Takens' embeddings were used to compute fractal dimensions in experimental data, identify the geometric operations in phase space behind the data, as well as to characterize the topological organization of the orbits, with its applications to model validation. Recently, very successful computational methods have been developed to reconstruct nonlinear dynamical system from data (1-7), and autoencoder networks were proposed as a novel way to make an embedding of the data into a reduced space where the dynamics may be simply represented (8-11). Given the enormous success of deep neural networks in finding intricate structure in large data sets, we anticipate that autoencoder related methods will become the new standard in the field (19,20). Yet these should be used with care. In this paper we studied the flow reconstructed by autoencoders from time series data generated by a chaotic system. We found that, even if this promising technique has the potential to generate embeddings of data, some of the flows reconstructed by neural networks trained with standard learning algorithms and metrics present the wrong topology. This raises a warning on the use of latent spaces as proper embeddings.

Availability of Data

The data that support the findings of this study are available from the corresponding author upon reasonable request

Acknowledgements

Grants from ANCyT (Argentina) and UBACyT (Universidad de Buenos Aires) are acknowledged. Data and codes are available upon request.

References

- 1) Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *science*, 324(5923), 81-85.
- 2) Wang, W. X., Yang, R., Lai, Y. C., Kovanis, V., & Grebogi, C. (2011). Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical review letters*, 106(15), 154101.
- 3) Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15), 3932-3937.
- 4) Wang, W. X., Lai, Y. C., & Grebogi, C. (2016). Data based identification and prediction of nonlinear and complex dynamical systems. *Physics Reports*, 644, 1-76.
- 5) Tran, G., & Ward, R. (2017). Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling & Simulation*, 15(3), 1108-1129.
- 6) Ding, J., Tarokh, V., & Yang, Y. (2018). Model selection techniques: An overview. *IEEE Signal Processing Magazine*, 35(6), 16-34.
- 7) AlMomani, A. A. R., Sun, J., & Bollt, E. (2020). How entropic regression beats the outliers problem in nonlinear system identification. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1), 013107.
- 8) Lusch, B., Kutz, J. N., & Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1), 1-10.
- 9) Champion, K., Lusch, B., Kutz, J. N., & Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45), 22445-22451.
- 10) Otto, S. E., & Rowley, C. W. (2019). Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1), 558-593.
- 11) Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52, 477-508.
- 12) Takens, F. (1981). Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980* (pp. 366-381). Springer, Berlin, Heidelberg.
- 13) Mindlin, G. B., Solari, H. G., Natiello, M. A., Gilmore, R., & Hou, X. J. (1991). Topological analysis of chaotic time series data from the Belousov-Zhabotinskii reaction. *Journal of Nonlinear Science*, 1(2), 147-173.

- 14) Mindlin, G. M., & Gilmore, R. (1992). Topological analysis and synthesis of chaotic time series. *Physica D: Nonlinear Phenomena*, 58(1-4), 229-242.
- 15) Brunton, S. L., & Noack, B. R. (2015). Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 67(5).
- 16) Duriez, T., Brunton, S. L., & Noack, B. R. (2017). Taming nonlinear dynamics with MLC. In *Machine Learning Control—Taming Nonlinear Dynamics and Turbulence* (pp. 93-120). Springer, Cham.
- 17) Berkooz G, Holmes P, Lumley JL (1993) The proper orthogonal decomposition in the analysis of turbulent flows. *Annu Rev Fluid Mech* 23:539–575
- 18) Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656, 5-28.
- 19) LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- 20) Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
- 21) Baldi, P., & Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1), 53-58.
- 22) Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.
- 23) Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, 184, 232-242.
- 24) Gilmore, R., & Lefranc, M. (2003). *The topology of chaos*. Weinheim, Germany: Wiley-VCH.
- 25) Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315-323).
- 26) Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- 27) Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256).
- 28) Our software implementation to compute linking numbers is publicly available at https://github.com/gon-uri/linking_number
- 29) Eckmann, J. P., Kamphorst, S. O., Ruelle, D., & Ciliberto, S. (1986). Liapunov exponents from time series. *Physical Review A*, 34(6), 4971.

- 30) Eckmann, J. P., & Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. In *The theory of chaotic attractors* (pp. 273-312). Springer, New York, NY.
- 31) Schölzel, Christopher. (2019, June 16). Nonlinear measures for dynamical systems (Version 0.5.2). Zenodo. <http://doi.org/10.5281/zenodo.3814723>
- 32) Mindlin, G. B., & Solari, H. G. (1995). Topologically inequivalent embeddings. *Physical Review E*, 52(2), 1497.

Figures

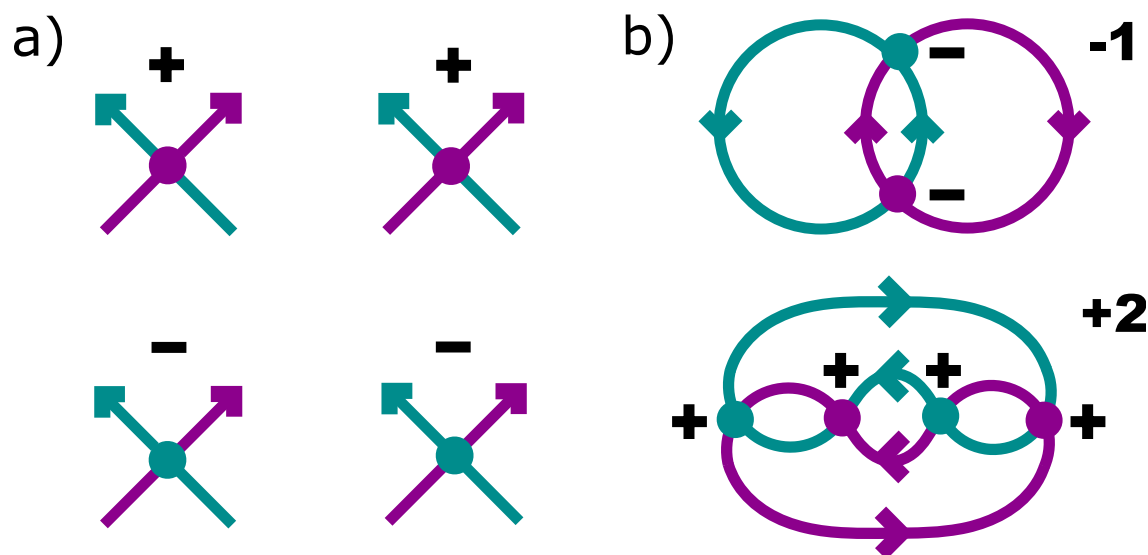


Figure 1

Linking Number computation between two three dimensional oriented curves. a) The four possible types of intersections in the two-dimensions projected curves. The colour of the circle at the intersection indicates which is the upper curve, while the sign indicates whether the intersection contributes $+1/2$ or $-1/2$ to the linking number. b) Examples of two pairs of curves, one with linking number -1 and the other with $+2$.

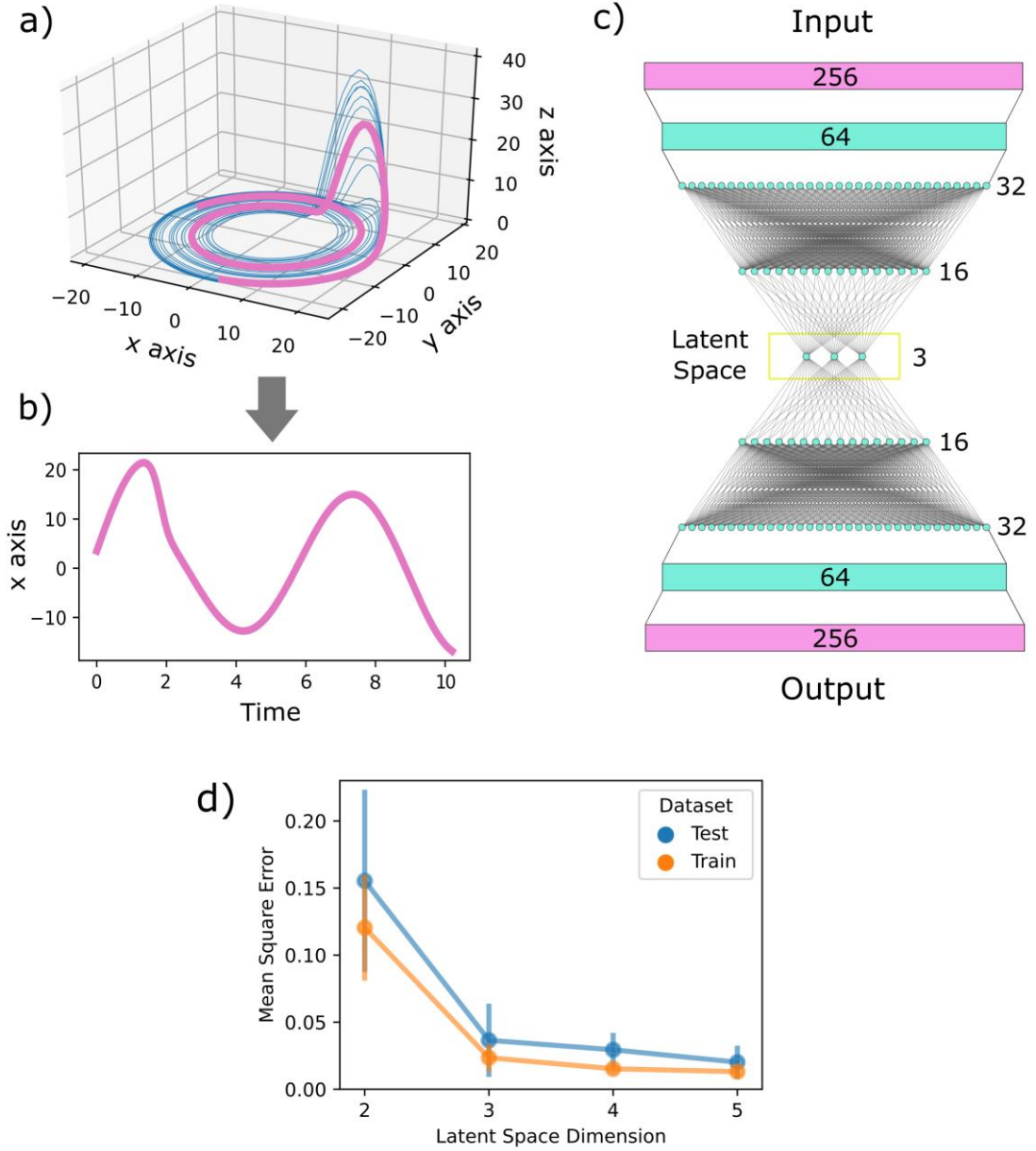


Figure 2

Autoencoders generate embeddings a) We simulated a strange attractor using a Rössler dynamical system. b) Fragments of 256 points were used to train a neural network. c) The architecture of our nine level layer. The cost function being minimized is the Mean Square Error between the input and output layers. d) The dimensionality of the middle layer was chosen as the minimum one that guarantees a qualitative gain in terms of MSE reduction. With a layer of two units, the MSE is an order of magnitude larger than with $N_{middle} \geq 2$.

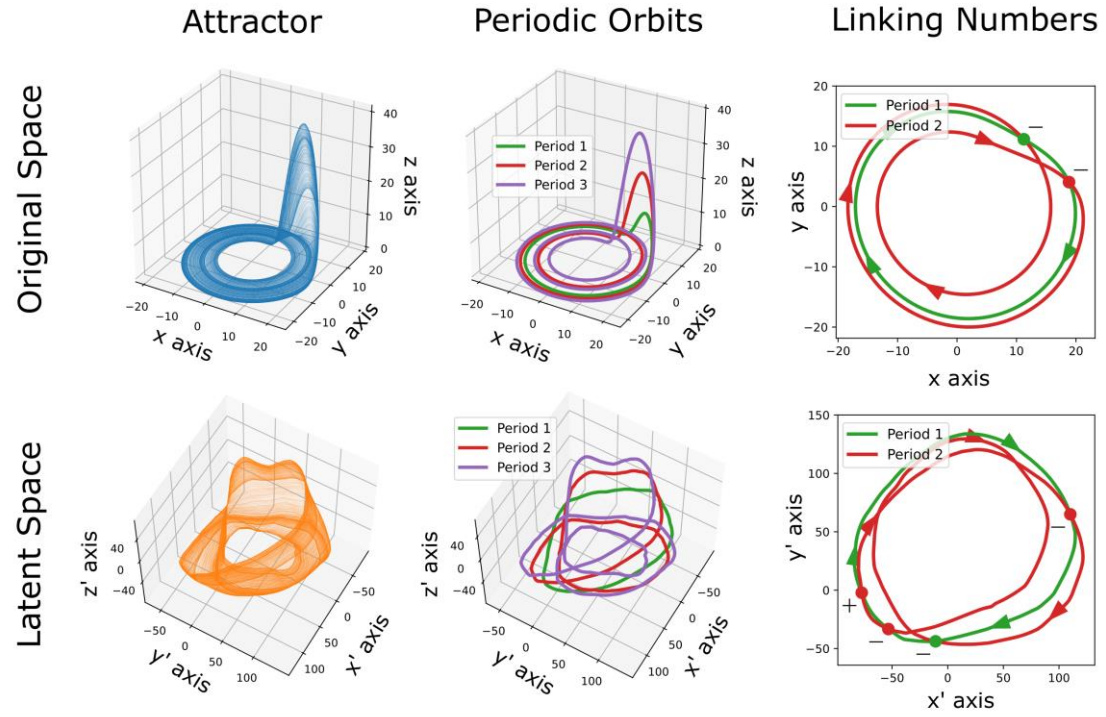


Figure 3

Comparison between the original flow, and the one reconstructed with the autoencoder. In the first column we show the original attractor and the reconstructed one. In the second column, we display the approximations of the periodic orbits in the original flow, and in the reconstructed ones. The third column displays the computation of the linking numbers for the cases of period 1 and 2. Both for the original and the reconstructed flows, $Linking(P1, P2) = -1$.

Figure 4
The results of training 60 autoencoders a) Proportion of reconstructed attractors with the right topological organization, for different epochs of the training. b) Evolution of the Mean Square Error during the training, discriminated by their topological structure. The points indicate the mean MSE, and the bars, their standard deviation. c) Example of a training process. At the top, we show the evolution of the MSE for the training and test sets. At the bottom, the linking numbers of the orbits in the reconstructed flow. The red indicates a topological organization different from that of the original flow. The green cells correspond to topological indexes as in the original flow.

Tables

					First Lyapunov Exponent		
Parameters			Number of Autoencoders		Original System	Latent Space	
a	b	c	Correct Topology	Incorrect Topology		Correct Topology	Incorrect Topology
0.1	0.1	14	53	7	0.072	0.076 (0.072 - 0.094)	0.09 (0.081 - 0.208)
0.12	0.12	9	17	3	0.068	0.069 (0.067 - 0.084)	0.074 (0.073 - 0.087)
0.11	0.11	13	16	4	0.086	0.093 (0.088 - 0.121)	0.248 (0.22 - 0.302)

Table 1

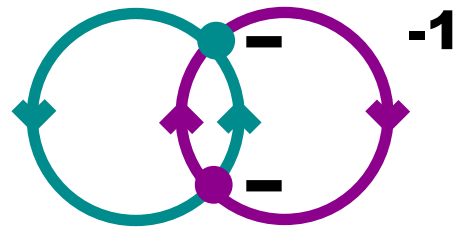
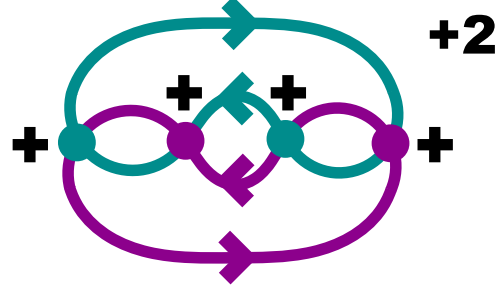
Values of the first Lyapunov exponent for the original and reconstructed attractors of Rössler systems with different sets of parameters. Values are reported as median (interquartile range).

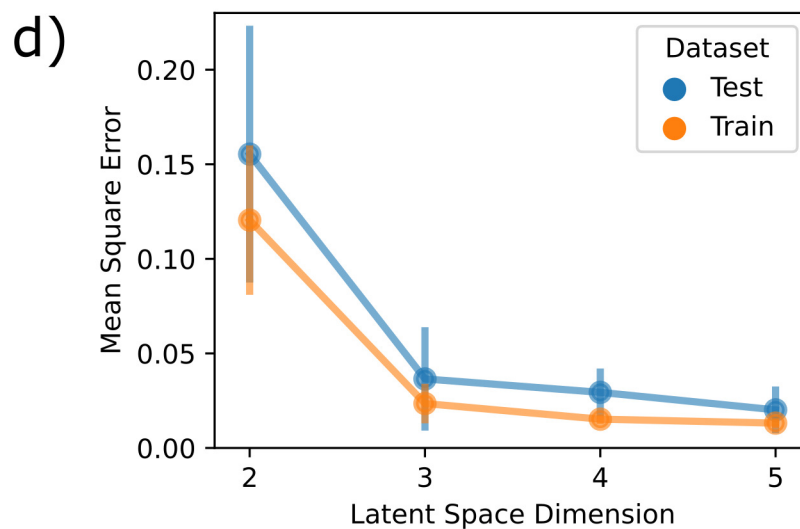
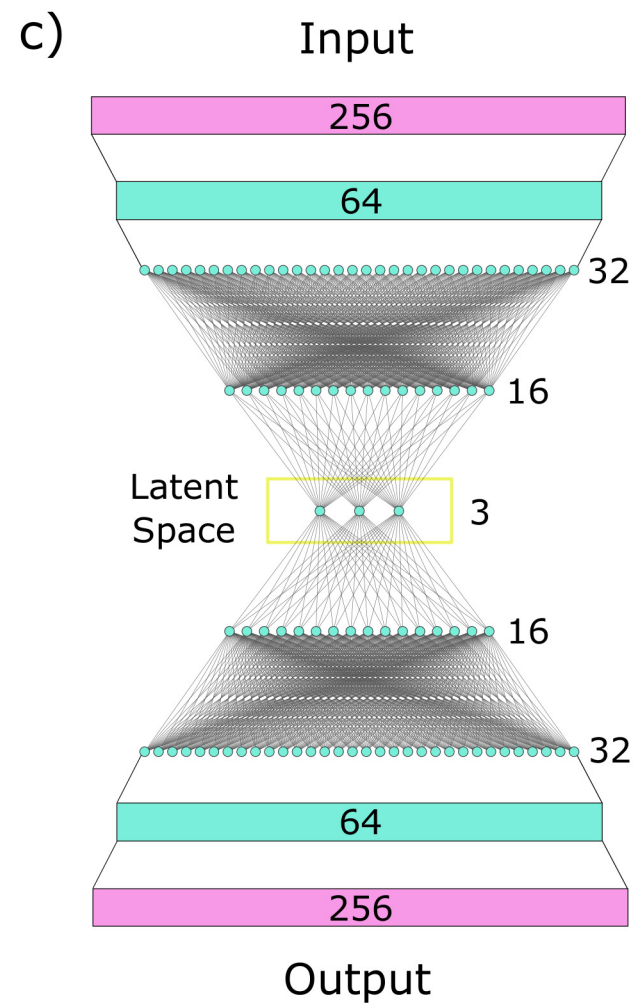
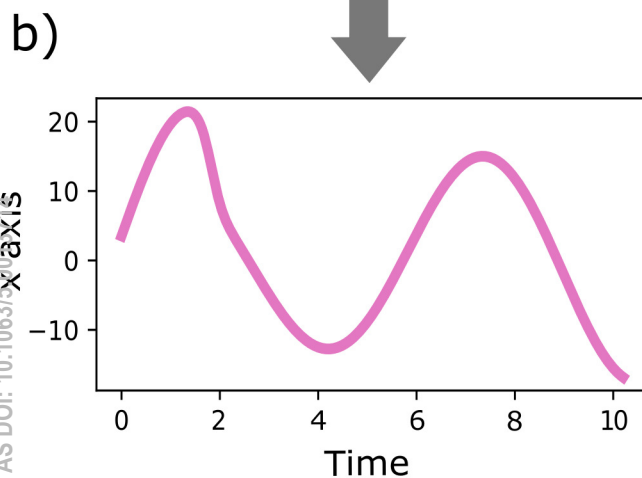
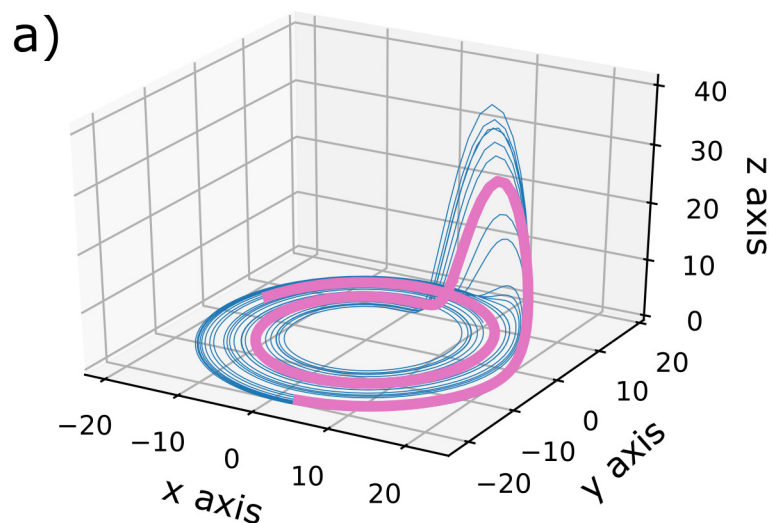
This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.
PLEASE CITE THIS ARTICLE AS DOI: 10.1063/5.0013714

a)



b)





This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1063/1.5015015

Original Space

