

Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics

Christoph Wehmeyer^{1, a)} and Frank Noé^{1, b)}

Freie Universität Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany

Inspired by the success of deep learning techniques in the physical and chemical sciences, we apply a modification of an autoencoder type deep neural network to the task of dimension reduction of molecular dynamics data. We can show that our time-lagged autoencoder reliably finds low-dimensional embeddings for high-dimensional feature spaces which capture the slow dynamics of the underlying stochastic processes—beyond the capabilities of linear dimension reduction techniques.

Molecular dynamics (MD) simulation allows us to probe the full spatiotemporal detail of molecular processes, but its usefulness has long been limited by the sampling problem. Recently, the combination of hard- and software for high-throughput MD simulations^{1–5} with Markov state models (MSM)^{6–8} has enabled the exhaustive statistical description of protein folding^{9–11}, conformational changes^{12,13}, protein-ligand association^{14–16} and even protein-protein association¹⁷. Using multi-ensemble Markov models (MEMMs)^{18–21}, even the kinetics of ultra-rare events beyond the seconds timescale are now available at atomistic resolution^{22–24}. A critical step in Markov state modeling and MSM-based sampling is the drastic dimension reduction from molecular configuration space to a space containing the slow collective variables (CVs)^{25–27}.

Another area that has made recent breakthroughs is deep learning, with impressive success in a variety of applications^{28,29} that indicate the capabilities of deep neural networks to uncover hidden structures in complex datasets. More recently, machine learning has also been successfully applied to chemical physics problems, such as learning quantum-chemical potentials, data-driven molecular design and binding site prediction^{30–35}. In the present study, we demonstrate that a deep time-lagged autoencoder network^{36,37} can be employed to perform the drastic dimension reduction required and find slow CVs that are suitable to build highly accurate MSMs.

Identification of slow CVs, sometimes called **reaction coordinates**, is an active area of research^{38–46}. State of the art methods for identifying slow CVs for MD are based on the variational approach for conformation dynamics (VAC)^{47,48} and its recent extension to non-equilibrium processes⁴⁹, which provide a systematic framework for finding the optimal slow CVs for a given time series. A direct consequence of the VAC is that time-lagged independent component analysis (TICA)^{50,51}, originally developed for blind-source separation^{52,53}, approximate the optimal slow CVs by linear combinations of molecular coordinates. An very similar method, developed in the dynamical systems community is dynamic

mode decomposition (DMD)^{54–56}. VAC and DMD find slow CVs *via* two different optimization goals using the time series $\{\mathbf{z}_t\}$:

1. *Variational approach* (VAC): search the d orthogonal directions \mathbf{r}_i , $i = 1, \dots, d$, such that the time-lagged autocorrelation of the projection $\mathbf{r}_i^\top \mathbf{z}_t$ is maximal⁵². These autocorrelations are bounded from above by the eigenvalues of the Markov propagator⁴⁷.
2. *Regression approach* (DMD): find the linear model \mathbf{K} with the minimal regression error $\sum_t \|\mathbf{z}_{t+\tau} - \mathbf{K}^\top \mathbf{z}_t\|^2$ and compute its d eigenvectors \mathbf{r}_i with largest eigenvalues.

Both approaches will give us the same directions \mathbf{r}_i if the corresponding sets of eigenvectors are used⁵⁷. These directions can be used for dimension reduction. By experience, we know that the dimension reduction can be made much more efficient by working in feature space instead of directly using the Cartesian coordinates^{38,39,42,47,48,58–63}. That means we perform some nonlinear mapping:

$$\mathbf{e}_t = E(\mathbf{z}_t), \quad (1)$$

e.g., by computing distances between residues or torsion angles, and then perform TICA or DMD (then known as EDMD⁶⁴) in the \mathbf{e}_t coordinates. Indeed this approach is fully described by the VAC⁴⁷ and will provide an optimal approximation of the slow components of the dynamics *via* a linear combination of the feature functions. If we do not want to choose the library of feature functions by hand, but instead want to optimize the nonlinear mapping E by employing a neural network, we have again two options: (1) employ the variational approach. This leads to VAMPnets described in⁶⁵, or (2) minimize the regression error:

$$\min_{D,E} \sum_t \|\mathbf{z}_{t+\tau} - D(E(\mathbf{z}_t))\|^2, \quad (2)$$

where D is some mapping from feature space to coordinate space and also includes the time-propagation. In this paper we investigate option (2), which naturally leads to using a time-lagged autoencoder (TAE).

An autoencoder (Fig. 1), is a type of deep neural network which is trained in a self-supervised manner^{36,37}.

^{a)}Electronic mail: christoph.wehmeyer@fu-berlin.de

^{b)}Electronic mail: frank.noel@fu-berlin.de

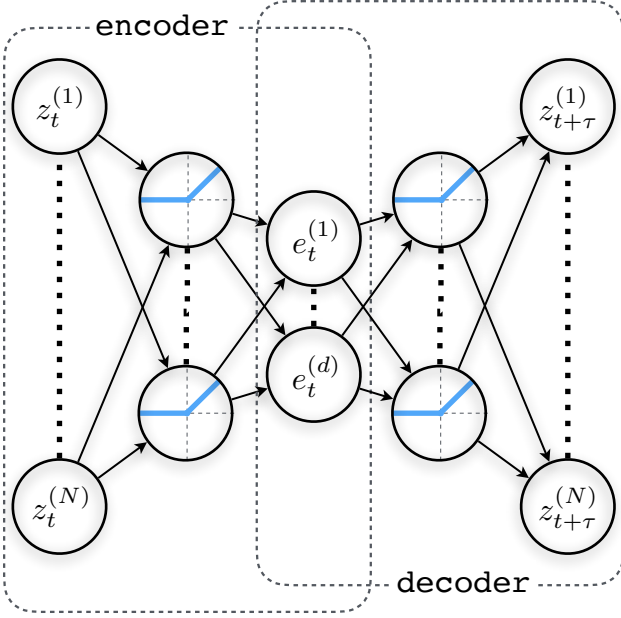


Figure 1. Schematic of a time-lagged autoencoder type neural network with one nonlinear hidden layer in each the encoding and decoding part. The encoder transforms a vector $\mathbf{z}_t \in \mathbb{R}^N$ to a d -dimensional latent space while the decoder maps this latent vector \mathbf{e}_t to the vector $\mathbf{z}_{t+\tau} \in \mathbb{R}^N$ in full coordinate space but at time τ later. For $\tau = 0$, this setup corresponds to a regular autoencoder.

The layer structure of the network is usually symmetric with a bottleneck in the middle and we refer to the first half including the bottleneck as the encoder while the second half is called decoder. Such a network is then trained to reconstruct its actual input \mathbf{z}_t with minimal regression error, i.e., the network must learn to encode an N -dimensional vector as a d -dimensional representation to pass the information through the bottleneck and reconstruct the original signal again in the decoder. Autoencoders can be viewed as a nonlinear version of a rank- d principal component analysis (PCA), and one can show that a linear autoencoder with bottleneck size d will identify the space of the d largest principal components⁶⁶. Autoencoders have been successfully applied to de-noise images^{67,68} and reduce the dimensionality of molecular conformations⁶⁹.

In this study, we alter the self-supervised training in such a way that the network minimizes the DMD regression error defined in Eq. (2), i.e., instead of training the network to reconstruct its input ($\mathbf{z}_t \mapsto \mathbf{z}_t$), we train it to predict a later frame ($\mathbf{z}_t \mapsto \mathbf{z}_{t+\tau}$); this is still self-supervised but requires to train on time series. While there are very strong mathematical arguments for employing the variational approach to learn nonlinear feature transformations *via* VAMPnets (see discussion in^{49,65}), there are practical arguments why the regression approach taken in the present TAEs is attractive: (i) we do not only learn the encoder network that performs the

dimension reduction, but we also learn the decoder network that can predict samples in our original coordinate space from points in the latent space, and (ii) TAEs can be extended towards powerful sampling methods such as variational and adversarial autoencoders^{70,71}. Here, we demonstrate that deep TAEs perform equally well or better than state of the art methods for finding the slow CVs in stochastic dynamical systems and biomolecules.

I. THEORY

To motivate our approach we first consider the case of linear transformations. We show that in a similar way that a linear autoencoder and PCA are equivalent up to orthogonalization, linear TAEs are equivalent to time-lagged canonical correlation analysis (TCCA), and in the time-reversible case equivalent to TICA. Then we move on to employ nonlinear TAEs for dimension reduction.

We are given a time-series with N dimensions and T time-steps, $\{\mathbf{z}_t \in \mathbb{R}^N\}_{t=1}^T$, and search for a d -dimensional embedding ($d < N$) which is well suited to compress the time-lagged data. To this aim, we define an encoding $E : \mathbb{R}^N \rightarrow \mathbb{R}^d$ and a decoding $D : \mathbb{R}^d \rightarrow \mathbb{R}^N$ operation which approximately reconstruct the time-lagged signal such that, on average, the error

$$\epsilon_t = \mathbf{z}_{t+\tau} - D(E(\mathbf{z}_t))$$

is small in some suitable norm. We introduce two conventions:

1. we employ mean-free coordinates,

$$\mathbf{x}_t = \mathbf{z}_t - \frac{1}{T-\tau} \sum_{s=1}^{T-\tau} \mathbf{z}_s$$

$$\mathbf{y}_t = \mathbf{z}_{t+\tau} - \frac{1}{T-\tau} \sum_{s=1}^{T-\tau} \mathbf{z}_{s+\tau},$$

2. and whiten them,

$$\tilde{\mathbf{x}}_t = \mathbf{C}_{00}^{-\frac{1}{2}} \mathbf{x}_t$$

$$\tilde{\mathbf{y}}_t = \mathbf{C}_{\tau\tau}^{-\frac{1}{2}} \mathbf{y}_t,$$

using the covariance matrices

$$\mathbf{C}_{00} = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \mathbf{x}_t \mathbf{x}_t^\top \quad (3)$$

$$\mathbf{C}_{0\tau} = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \mathbf{x}_t \mathbf{y}_t^\top \quad (4)$$

$$\mathbf{C}_{\tau\tau} = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \mathbf{y}_t \mathbf{y}_t^\top. \quad (5)$$

Note that whitening must take into account that \mathbf{C}_{00} and $\mathbf{C}_{\tau\tau}$ are often not full rank matrices⁷².

Now we must find an encoding and decoding which minimizes the reconstruction error

$$\min_{E,D} \sum_{t=1}^{T-\tau} \|\tilde{\mathbf{y}}_t - D(E(\tilde{\mathbf{x}}_t))\|_2^2 \quad (6)$$

for a selected class of functions E and D . The simplest choice, of course, are linear functions.

A. Linear TAE performs TCCA

As we operate on mean free data, we can represent linear encodings and decodings by simple matrix multiplications:

$$\begin{aligned} E(\tilde{\mathbf{x}}_t) &= \tilde{\mathbf{E}}\tilde{\mathbf{x}}_t \\ D(E(\tilde{\mathbf{x}}_t)) &= \tilde{\mathbf{D}}\tilde{\mathbf{E}}\tilde{\mathbf{x}}_t \end{aligned}$$

with the encoding matrix $\tilde{\mathbf{E}} \in \mathbb{R}^{d \times N}$, which projects N -dimensional data onto a d -dimensional space, and the decoding matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{N \times d}$, which lifts the encoded data back to an N -dimensional vector space.

For convenience, we define the matrices $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{T-\tau}]$ and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_{T-\tau}]$, and likewise $\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}$ for the whitened coordinates. The minimal reconstruction error (6) thus becomes

$$\min_{\tilde{\mathbf{K}}_d} \|\tilde{\mathbf{Y}} - \tilde{\mathbf{K}}_d \tilde{\mathbf{X}}\|_F^2, \quad (7)$$

where F denotes the Frobenius norm and $\tilde{\mathbf{K}}_d = \tilde{\mathbf{D}}\tilde{\mathbf{E}}$ is the rank- d Koopman matrix; for the full-rank Koopman matrix, we simply omit the rank index: $\tilde{\mathbf{K}}_N = \tilde{\mathbf{K}}$.

Eq. (7) is a linear least squares problem. The full-rank solution is given by the regression

$$\tilde{\mathbf{K}} = \tilde{\mathbf{Y}}\tilde{\mathbf{X}}^\top (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top)^{-1} = \frac{1}{T-\tau} \tilde{\mathbf{Y}}\tilde{\mathbf{X}}^\top,$$

where we have used the fact that the data is whitened: $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top = (T-\tau)\mathbf{I}$. Using the definition of the covariance matrices (3-5), we can also write

$$\tilde{\mathbf{K}}^\top = \tilde{\mathbf{X}}\tilde{\mathbf{Y}}^\top = \mathbf{C}_{00}^{-\frac{1}{2}} \mathbf{C}_{0\tau} \mathbf{C}_{\tau\tau}^{-\frac{1}{2}}.$$

This form is often referred to as half-weighted Koopman matrix and it arises naturally when using whitened data.

The last step to the solution lies in choosing the optimal rank- d approximation to the Koopman matrix which is given by the rank- d singular value decomposition,

$$\begin{aligned} \tilde{\mathbf{K}}_d^\top &= \text{svd}_d \left(\mathbf{C}_{00}^{-\frac{1}{2}} \mathbf{C}_{0\tau} \mathbf{C}_{\tau\tau}^{-\frac{1}{2}} \right) \\ &= \mathbf{U}_d \Sigma_d \mathbf{V}_d^\top, \end{aligned}$$

where d indicates that we take the d largest singular values and corresponding singular vectors. Thus, a possible choice for the encoding and decoding matrices for whitened data is

$$\begin{aligned} \tilde{\mathbf{E}} &= \Sigma_d \mathbf{V}_d^\top \\ \tilde{\mathbf{D}} &= \mathbf{U}_d. \end{aligned}$$

With this choice, we find for the mean-free but non-whitened data

$$\begin{aligned} \tilde{\mathbf{y}}_t &= \tilde{\mathbf{K}} \tilde{\mathbf{x}}_t \\ \Leftrightarrow \mathbf{C}_{\tau\tau}^{-\frac{1}{2}} \mathbf{y}_t &= \left(\mathbf{C}_{00}^{-\frac{1}{2}} \mathbf{C}_{0\tau} \mathbf{C}_{\tau\tau}^{-\frac{1}{2}} \right)^\top \mathbf{C}_{00}^{-\frac{1}{2}} \mathbf{x}_t \\ \Leftrightarrow \mathbf{y}_t &= \mathbf{C}_{0\tau}^\top \mathbf{C}_{00}^{-1} \mathbf{x}_t \end{aligned}$$

the non-whitened Koopman matrix consistently with^{47,49,64,73}:

$$\mathbf{K}^\top = \mathbf{C}_{00}^{-1} \mathbf{C}_{0\tau}.$$

Likewise, we find the non-whitened encoding and decoding matrices

$$\begin{aligned} \mathbf{E} &= \Sigma_d \mathbf{V}_d^\top \mathbf{C}_{00}^{-\frac{1}{2}} \\ \mathbf{D} &= \mathbf{C}_{\tau\tau}^{\frac{1}{2}} \mathbf{U}_d, \end{aligned}$$

where the encoding consists of a whitening followed by the whitened encoding, while the decoding starts with the whitened decoding followed by unwhitening. This solution is equivalent with time-lagged canonical correlation analysis (TCCA)^{49,74}.

B. Time-reversible linear TAE performs TICA

If the covariance matrix $\mathbf{C}_{0\tau}$ is symmetric, the singular value decomposition of the full-rank Koopman matrix is equivalent to an eigenvector decomposition:

$$\tilde{\mathbf{K}}_d = \mathbf{U}_d \Sigma_d \mathbf{U}_d^\top.$$

If we further have a stationary time series, i.e., $\mathbf{C}_{00} = \mathbf{C}_{\tau\tau}$, the non-whitened encoding and decoding matrices are

$$\begin{aligned} \mathbf{E} &= \Sigma_d \mathbf{U}_d^\top \mathbf{C}_{00}^{-\frac{1}{2}} \\ \mathbf{D} &= \mathbf{C}_{00}^{\frac{1}{2}} \mathbf{U}_d, \end{aligned}$$

where $\mathbf{U}_d^\top \mathbf{C}_{00}^{-\frac{1}{2}}$ contains the usual TICA eigenvectors and multiplication with Σ_d transforms to a kinetic map. Thus, if we include Σ_d in the decoder part, this solution is equivalent to TICA⁵⁰⁻⁵², while if we include it in the encoder, it is equivalent to a kinetic map⁷⁵.

Motivated by these theoretical results we will employ TAEs to learn nonlinear encodings and decodings that optimize Eq. (6).

II. EXPERIMENTS

We put the nonlinear time-lagged autoencoder to the test by applying it to two toy models of different degree of difficulty as well as molecular dynamics data for alanine dipeptide. In all three cases, we compare the performance of the autoencoder with that of TICA (with kinetic map scaling) and PCA by

1. comparing the reconstruction errors (6) of the validation sets,
2. comparing the low-dimensional representations found with the known essential variables of the respective system by employing canonical correlation analysis (CCA), and
3. examining the suitability of the encoded space for building MSMs via convergence of implied timescales.

The time-lagged autoencoders used in this study are implemented using the PyTorch framework⁷⁶ and consist of an input layer with N units, followed by one or two hidden layers with sizes H_1 and H_2 and the latent layer with size d which concludes the encoding stage. The decoding part also adds one or two hidden layers of the same sizes as in the encoding part, followed by the output layer with size N . All hidden layers employ leaky rectified linear units⁷⁷ (leaky parameter $\alpha = 0.001$) and a dropout layer⁷⁸ (dropout probability $p = 0.5$). We train the networks using the Adam⁷⁹ optimizer.

To account for the stochastic nature of the high dimensional data, the autoencoder training process, and the discretization when building MSMs, all simulations have been repeated 100 times while shuffling training and validation sets. We show the ensemble median as well as a one-standard-deviation percentile (68%). The evaluation process always follows the pattern

1. Gather the high dimensional data and reference low-dimensional representation *via* independent simulation or bootstrapping.
2. Train the encoder/decoder for all techniques on two thirds (training set) of the high dimensional data.
3. Compute the reconstruction error for the remaining third of the data (validation set).
4. Obtain encoded coordinates and whiten (training + validation sets).
5. Perform CCA to compare the encoded space to the reference data (training + validation sets).
6. Build MSMs⁸⁰ on the encoded space (training + validation sets) and validate using implied timescales tests⁸¹.

A. Two-state toy model

The first toy model is based on a two-state hidden Markov model (HMM) which emits anisotropic Gaussian noise in the two-dimensional x/y -plane. To complicate matters we perform the operation

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y + \sqrt{x} \end{pmatrix}$$

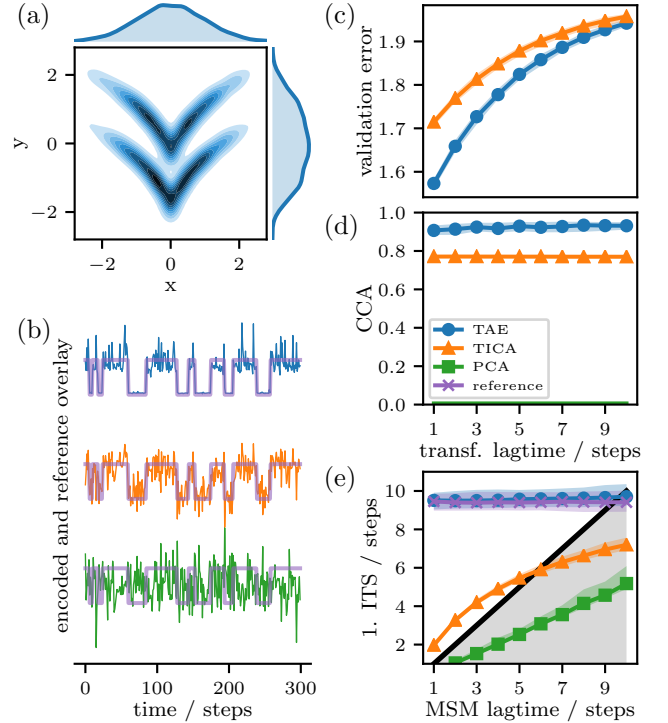


Figure 2. **Two-dimensional two-state system that is not linearly separable.** (a) Joint and marginal distributions of the two-state HMM toy model. (b) Comparison of time-series segments of one-dimensional transformations (lag time $\tau = 1$ step) with the actual hidden state time series. (c) Regression error for the validation set. (d) Canonical correlation coefficient between encoded time series and true hidden state time-series as a function of the transformation lagtime. (e) Convergence of the slowest implied timescale for the one-dimensional transformations and the hidden state time series. Panels (c-e) show the median (lines) and 68% percentiles (shaded areas) over 100 independent realizations.

which leads to the distribution shown in Fig. 2a. We compare one-dimensional representations found by applying TICA and PCA to the time series, and a TAE employing one hidden layer of 50 units in the encoding and decoding part each, and a bottleneck size of $d = 1$.

The TAE-encoded variable overlaps very well with the hidden state time series and can clearly separate both hidden states, while TICA gives a more blurred picture with no clear separation and PCA does not seem to separate the hidden states at all (Fig. 2b). These differences are quantified by the CCA score between the encoded and true hidden state signals (Fig. 2d). The time-lagged autoencoder outperforms TICA at all examined transformation lagtimes in terms of the reconstruction error; the difference is particularly strong for small lagtimes (Fig. 2c). Finally, the encoding found by the TAE is excellently suited to build an MSM that approximates the slowest relaxation timescale even at short lagtimes (Fig. 2e). In contrast, the MSM based on TICA converges towards the true timescale too slowly, and does not get close to it be-

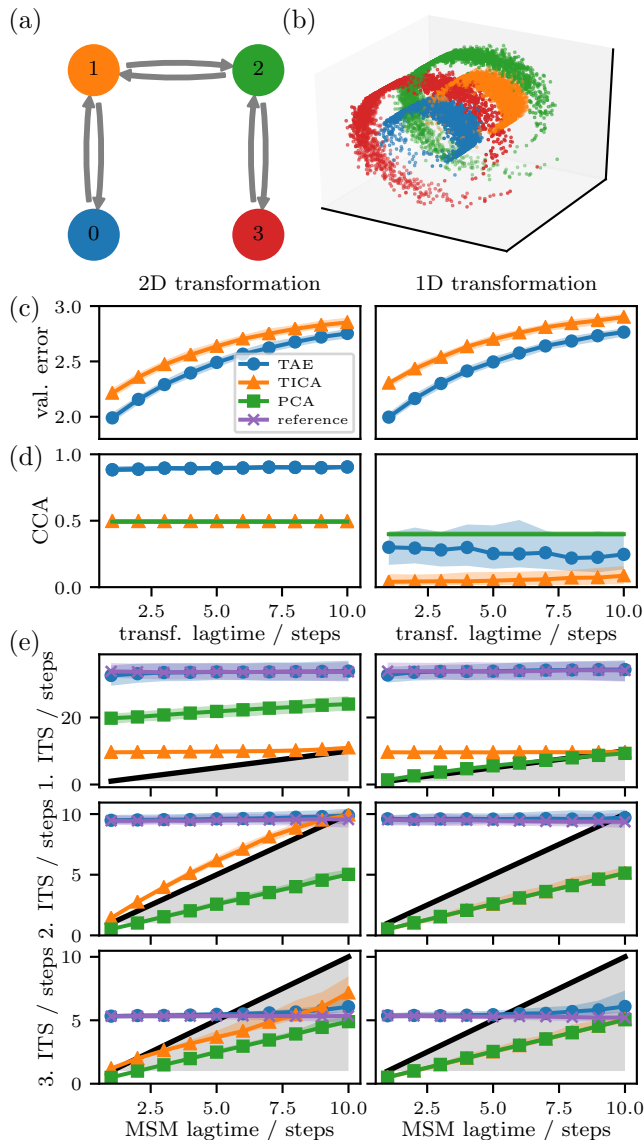


Figure 3. **Three-dimensional four-state system that is not linearly separable.** (a) Network of the four-state HMM toy model. (b) Emissions in a three-dimensional swissroll shape. (c) Regression error of the validation set. (d) Canonical correlation between the true HMM time series with the one- or two-dimensional encoding, as a function of the lagtime τ . (e) The first three implied timescales (ITS) obtained from MSMs constructed on the encoding space. The TAE and TICA were trained at a lagtime of one simulation step. All lines show the median over 100 realizations, shaded areas indicate 68% percentiles.

fore reaching the numerically invalid range $\tau > t_2$. The MSM build on PCA seems to be completely unsuitable for recovering kinetics (Fig. 2e).

B. Four-state swissroll toy model

The second toy model is based on a four-state hidden Markov model (HMM), which emits isotropic Gaussian noise in the two-dimensional x/y -plane, with the means of the states located as shown in Fig. 3a. To create a nonlinearly separable system, we perform the operation

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \cos(x) \\ y \\ x \sin(x) \end{pmatrix}$$

which produces a picture that is reminiscent of the swiss roll commonly used as a benchmark for nonlinear dimension reduction (Fig. 3b). For this toy model, we examine two- and one-dimensional encodings. The TAE with $d = 2$ uses a single hidden layer with 100 units in the encoder and decoder part, while the TAE with $d = 1$ uses two hidden layers with 200 and 100 units in the encoder part and 100 and 200 units in the decoder.

In both cases, the time-lagged autoencoder outperforms TICA in terms of reconstruction error (Fig. 3c). Again, the difference is larger for small transformation lagtimes. Indeed, the TAE encoding is nearly perfectly correlated with the true hidden states time series (Fig. 3d), while both TICA and PCA are significantly worse and nearly identical to each other. In the one-dimensional case, all methods fail at obtaining a high correlation, indicating that this system is not perfectly separable with a single coordinate, even if it is nonlinear.

MSMs constructed on the encoded space also indicate that the TAE perfectly recovers the reference timescales at all lagtimes (Fig. 3e) – surprisingly this is also true for the one-dimensional embedding, despite the fact that this embedding is not well correlated with the true hidden time series. MSMs build on either the TICA or PCA space are systematically underestimated and mostly show no sign of convergence.

C. Molecular dynamics data of alanine dipeptide

Our third example involves real MD data from three independent simulations of 250 ns each^{82,83} from which we repeatedly bootstrap five sub-trajectories of length 100 ns. The features on which we apply the encoding are RMSD-aligned heavy atom positions which yield an $N = 30$ -dimensional input space. Although we do not know the optimal two-dimensional representation, we assume that the commonly used (ϕ, ψ) backbone dihedrals contain all the relevant long-time behavior of the system (except for methyl rotations which do not affect the heavy atoms⁸⁴), and we thus use these dihedral angles as a reference to compare our encoding spaces to. Fig. 4a and b show the free energy surface for the reference representation and the assignment of (ϕ, ψ) -points to the four most slowly-interconverting metastable states.

The TAE outperforms TICA in terms of the regression error (Fig. 4c). While all three methods find a two-

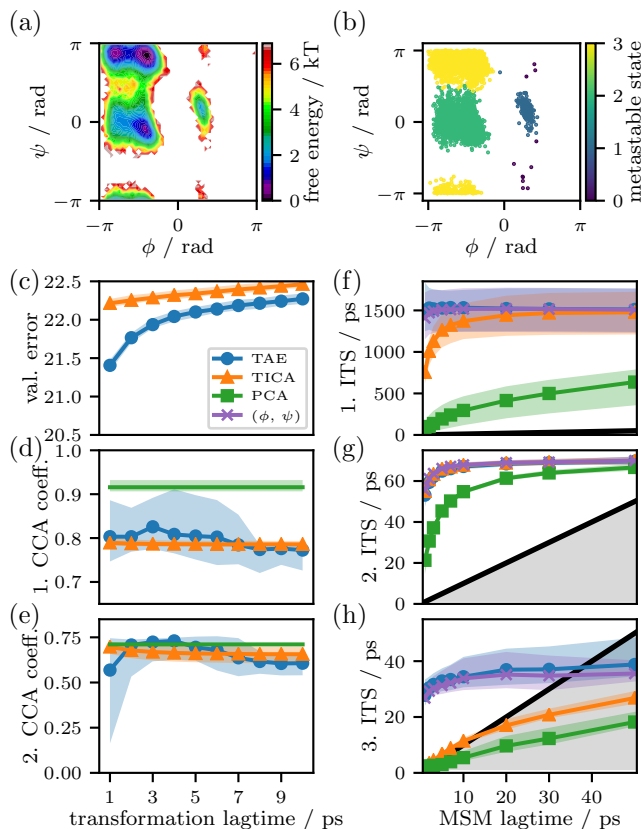


Figure 4. **Alanine dipeptide kinetics estimated from the time series in 30 heavy atom coordinates.** (a) Free energy surface in the well-known space of ϕ/ψ backbone dihedrals. (b) Metastable partition into the four most slowly-interconverting states. (c) Regression error of the validation set. (d, e) Canonical correlations between the ϕ/ψ dihedral representation and the two-dimensional encoded spaces found by TAE, TICA or PCA. (f-g) First three implied timescales (ITS) of MSMs constructed in the encoded space. The TAE was trained at lagtime of 3ps and TICA at a lagtime 1ps. All lines show the median over 100 bootstrapped samples, shaded areas indicate 68% percentiles.

dimensional space that correlates relatively well with the (ϕ, ψ) -plane, PCA achieves, surprisingly the best correlation (Fig. 4d-e), while the TAE and TICA are similar. This result is put into perspective by the performances of MSMs built upon the encoding space (Fig. 4f-h). Here, TAE clearly performs best. The TICA MSM does converge to the first two relaxation timescales, although slower than the TAE in the first relaxation timescale, while its convergence of the third relaxation timescale is too slow to be practically useful. PCA performs poorly for all relaxation timescales.

III. CONCLUSION

We have investigated the performance of a special type of deep neural network, the time-lagged autoencoder, to

the task of finding low-dimensional, nonlinear embeddings of dynamical data. We have first shown that a linear time-lagged autoencoder is equivalent to time-lagged canonical correlation analysis, and for the special case of statistically time-reversible data equivalent to the time-lagged independent component analysis commonly used in the analysis of MD data. However, in many datasets, the metastable states are not linearly separable and there is thus no low-dimensional linear subspace that will resolve the slow processes, resulting in large approximation errors of MSMs and other estimators of kinetics or thermodynamics. In these cases, the traditional variational approach puts the workload on the user who can mitigate this problem by finding suitable feature transformations of the MD coordinates, e.g., to contact maps, distances, angles or other other nonlinear functions in which the metastable states may be linearly separable. In a deep TAE, instead, we take the perspective that the nonlinear feature transformation should be found automatically by an optimization algorithm. Our results on toy models and MD data indicate that this is indeed possible and low-dimensional representations can be found that outperform those found by naive TICA and PCA.

Our approach is closely related to the previously proposed VAMPnet approach that performs a simultaneous dimension reduction and MSM estimation by employing the variational approach of Markov processes⁶⁵. By combining the theoretical results from this paper with those of^{47,49}, it is clear that in the linear case all these methods are equivalent with TCCA, TICA, Koopman models or MSMs, depending on the type of inputs used, and whether the data are reversible or nonreversible. We believe that there is also a deeper mathematical relationship between these methods in the nonlinear case, e.g., when deep neural networks are employed to learn the feature transformation, but this relationship is still elusive. Both the present approach, that minimizes the TAE regression error in the input space, as well as the variational approach, that maximizes a variational score in the feature space^{47,49}, are suitable to conduct hyper-parameter search⁸⁵. The present error model (6) is based on least square regression, or in other words, on the assumption of additive noise in the configuration space, while VAMPnets do not have this restriction. Also, VAMPnets can incorporate the MSM estimation in a single end-to-end learning framework. On the other hand, the autoencoder approach has the advantage that, in addition to the feature encoding, a feature decoding back to the full configuration space is learned, too. Future studies will investigate the strengths and weaknesses of both approaches in greater detail.

ACKNOWLEDGMENTS

We are grateful for insightful discussions with Steve Brunton, Nathan Kutz, Andreas Mardt, Luca Pasquali, and Simon Olsson. We gratefully acknowledge funding by

European Commission (ERC StG 307494 “pcCell”) and Deutsche Forschungsgemeinschaft (SFB 1114/A04).

REFERENCES

- ¹M. Shirts and V. S. Pande, *Science* **290**, 1903 (2000).
- ²I. Buch, M. J. Harvey, T. Giorgino, D. P. Anderson, and G. De Fabritiis, *J. Chem. Inf. Model.* **50**, 397 (2010).
- ³D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. Dror, M. Eastwood, J. Bank, J. Jumper, J. Salmon, Y. Shan, and W. Wriggers, *Science* **330**, 341 (2010).
- ⁴S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, B. Hess, and E. Lindahl, *Bioinformatics* **29**, 845 (2013).
- ⁵S. Doerr, M. J. Harvey, F. Noé, and G. D. Fabritiis, *J. Chem. Theory Comput.* **12**, 1845 (2016).
- ⁶J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, *J. Chem. Phys.* **134**, 174105 (2011).
- ⁷G. R. Bowman, V. S. Pande, and F. Noé, eds., *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation* (Springer Netherlands, 2014).
- ⁸M. Sarich and C. Schütte, *Metastability and Markov State Models in Molecular Dynamics*, Courant Lecture Notes (American Mathematical Society, 2013).
- ⁹F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl, *Proc. Natl. Acad. Sci. USA* **106**, 19011 (2009).
- ¹⁰G. R. Bowman, K. A. Beauchamp, G. Boxer, and V. S. Pande, *J. Chem. Phys.* **131**, 124101 (2009).
- ¹¹K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, *Science* **334**, 517 (2011).
- ¹²S. K. Sadiq, F. Noé, and G. De Fabritiis, *Proc. Natl. Acad. Sci. USA* **109**, 20449 (2012).
- ¹³K. J. Kohlhoff, D. Shukla, M. Lawrenz, G. R. Bowman, D. E. Konerding, D. Belov, R. B. Altman, and V. S. Pande, *Nat. Chem.* **6**, 15 (2014).
- ¹⁴I. Buch, T. Giorgino, and G. De Fabritiis, *Proc. Natl. Acad. Sci. USA* **108**, 10184 (2011).
- ¹⁵D.-A. Silva, G. R. Bowman, A. Sosa-Peinado, and X. Huang, *PLoS Comput. Biol.* **7**, e1002054 (2011).
- ¹⁶N. Plattner and F. Noé, *Nat. Commun.* **6**, 7653 (2015).
- ¹⁷N. Plattner, S. Doerr, G. D. Fabritiis, and F. Noé, *Nat. Chem.* **9**, 1005 (2017).
- ¹⁸H. Wu, A. S. J. S. Mey, E. Rosta, and F. Noé, *J. Chem. Phys.* **141**, 214106 (2014).
- ¹⁹E. Rosta and G. Hummer, *J. Chem. Theory Comput.* **11**, 276 (2015).
- ²⁰H. Wu, F. Paul, C. Wehmeyer, and F. Noé, *Proc. Natl. Acad. Sci. USA* **113**, E3221 (2016).
- ²¹A. S. J. S. Mey, H. Wu, and F. Noé, *Phys. Rev. X* **4**, 041018 (2014).
- ²²F. Paul, C. Wehmeyer, E. T. Abualrous, H. Wu, M. D. Crabtree, J. Schöneberg, J. Clarke, C. Freund, T. R. Weikl, and F. Noé, *Nat. Commun.* **8** (2017).
- ²³R. Casasnovas, V. Limongelli, P. Tiwary, P. Carloni, and M. Parrinello, *J. Am. Chem. Soc.* **139**, 4780 (2017).
- ²⁴P. Tiwary, J. Mondal, and B. J. Berne, *Sci. Adv.* **3**, e1700014 (2017).
- ²⁵F. Noé and C. Clementi, *Curr. Opin. Struc. Biol.* **43**, 141 (2017).
- ²⁶J. Preto and C. Clementi, *Phys. Chem. Chem. Phys.* **16**, 19181 (2014).
- ²⁷J. McCarty and M. Parrinello, [arXiv:1703.08777](https://arxiv.org/abs/1703.08777).
- ²⁸Y. LeCun, Y. Bengio, and G. Hinton, *Nature* **521**, 436 (2015).
- ²⁹D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, and et al., *Nature* **529**, 484 (2016).
- ³⁰M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, *Phys. Rev. Lett.* **108** (2012).
- ³¹R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, *ArXiv e-prints* (2016), [arXiv:1610.02415](https://arxiv.org/abs/1610.02415) [cs.LG].
- ³²G. B. Goh, N. O. Hodas, and A. Vishnu, *J. Comp. Chem.* **38**, 1291 (2017).
- ³³K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, *Nat. Commun.* **8**, 13890 (2017).
- ³⁴E. Schneider, L. Dai, R. Q. Topper, C. Drechsel-Grau, and M. E. Tuckerman, *Phys. Rev. Lett.* **119** (2017).
- ³⁵J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose, and G. De Fabritiis, *Bioinformatics* **33**, 3036 (2017).
- ³⁶D. E. Rumelhart, G. E. Hinton, and R. J. Williams (MIT Press, Cambridge, MA, USA, 1986) Chap. Learning Internal Representations by Error Propagation, pp. 318–362.
- ³⁷P. Baldi, in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Proceedings of Machine Learning Research, Vol. 27, edited by I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver (PMLR, Bellevue, Washington, USA, 2012) pp. 37–49.
- ³⁸B. Peters and B. L. Trout, *J. Chem. Phys.* **125**, 054108 (2006).
- ³⁹M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi, *J. Chem. Phys.* **134**, 124116 (2011).
- ⁴⁰M. A. Rohrdanz, W. Zheng, and C. Clementi, *Ann. Rev. Phys. Chem.* **64**, 295 (2013).
- ⁴¹H. Stamati, C. Clementi, and L. E. Kavraki, *Proteins* **78**, 223 (2010).
- ⁴²P. Das, M. Moll, H. Stamati, L. E. Kavraki, and C. Clementi, *Proc. Natl. Acad. Sci. USA* **103**, 9885 (2006).
- ⁴³B. Peters, *J. Chem. Phys.* **125**, 241101 (2006).
- ⁴⁴J.-H. Prinz, J. D. Chodera, and F. Noé, *Phys. Rev. X* **4**, 011020 (2014).
- ⁴⁵A. Altis, P. H. Nguyen, R. Hegger, and G. Stock, *J. Chem. Phys.* **126**, 244111 (2007).
- ⁴⁶S. V. Krivov and M. Karplus, *Proc. Nat. Acad. Sci. USA* **101**, 14766 (2004).
- ⁴⁷F. Noé and F. Nüske, *Multiscale Model. Simul.* **11**, 635 (2013).
- ⁴⁸F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, *J. Chem. Theory Comput.* **10**, 1739 (2014).
- ⁴⁹H. Wu and F. Noé, [arXiv:1707.04659](https://arxiv.org/abs/1707.04659) (2017).
- ⁵⁰G. Pérez-Hernández, F. Paul, T. Giorgino, G. D. Fabritiis, and F. Noé, *J. Chem. Phys.* **139**, 015102 (2013).
- ⁵¹C. R. Schwantes and V. S. Pande, *J. Chem. Theory Comput.* **9**, 2000 (2013).
- ⁵²L. Molgedey and H. G. Schuster, *Phys. Rev. Lett.* **72**, 3634 (1994).
- ⁵³A. Ziehe and K.-R. Müller, in *ICANN 98* (Springer Science and Business Media, 1998) pp. 675–680.
- ⁵⁴I. Mezić, *Nonlinear Dynam.* **41**, 309 (2005).
- ⁵⁵P. J. Schmid and J. Sesterhenn, in *61st Annual Meeting of the APS Division of Fluid Dynamics. American Physical Society* (2008).
- ⁵⁶J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *J. Comput. Dyn.* **1**, 391 (2014).
- ⁵⁷S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, [arXiv:1703.10112](https://arxiv.org/abs/1703.10112) (2017).
- ⁵⁸S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller, *Neur. Comp.* **15**, 1089 (2003).
- ⁵⁹C. R. Schwantes and V. S. Pande, *J. Chem. Theory Comput.* **11**, 600 (2015).
- ⁶⁰M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, [arXiv:1411.2260](https://arxiv.org/abs/1411.2260) (2014).
- ⁶¹F. Nüske, R. Schneider, F. Vitalini, and F. Noé, *J. Chem. Phys.* **144**, 054105 (2016).
- ⁶²S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Proc. Natl. Acad. Sci. USA* **113**, 3932.
- ⁶³M. P. Harrigan and V. S. Pande, [bioRxiv, 123752](https://arxiv.org/abs/123752) (2017).
- ⁶⁴M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, *J. Nonlinear Sci.* **25**, 1307 (2015).
- ⁶⁵A. Mardt, L. Pasquali, H. Wu, and F. Noé, *Nat. Commun.* , [arXiv:1710.06012](https://arxiv.org/abs/1710.06012) (in revision).

- ⁶⁶P. Baldi and K. Hornik, J. Neural Networks **2**, 53 (1989).
- ⁶⁷G. E. Hinton, Science **313**, 504 (2006).
- ⁶⁸Y. Wang, H. Yao, and S. Zhao, Neurocomputing **184**, 232 (2016).
- ⁶⁹W. M. Brown, S. Martin, S. N. Pollock, E. A. Coutsiias, and J.-P. Watson, J. Chem. Phys. **129**, 064118 (2008).
- ⁷⁰M. W. Diederik P Kingma, in *ICLR* (2014).
- ⁷¹A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, arXiv:1511.05644 (2016).
- ⁷²H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noé, J. Chem. Phys. **146**, 154104 (2017).
- ⁷³I. Horenko, C. Hartmann, C. Schütte, and F. Noé, Phys. Rev. E **76**, 016706 (2007).
- ⁷⁴H. Hotelling, Biometrika **28**, 321 (1936).
- ⁷⁵F. Noé and C. Clementi, J. Chem. Theory Comput. **11**, 5002 (2015).
- ⁷⁶A. Paszke, S. Gross, S. Chintala, and G. Chanan, “Tensors and dynamic neural networks in python with strong gpu acceleration,” <https://github.com/pytorch/pytorch> (2017).
- ⁷⁷A. L. Maas, A. Y. Hannun, and A. Y. Ng, in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (2013).
- ⁷⁸N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, J. Mach. Learn. Res. **15**, 1929 (2014).
- ⁷⁹D. P. Kingma and J. Ba, ArXiv e-prints (2014), arXiv:1412.6980v9 [cs.LG].
- ⁸⁰M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, J. Chem. Theory Comput. **11**, 5525 (2015).
- ⁸¹W. C. Swope, J. W. Pitera, and F. Suits, J. Phys. Chem. B **108**, 6571 (2004).
- ⁸²F. Nüske, H. Wu, J.-H. Prinz, C. Wehmeyer, C. Clementi, and F. Noé, J. Chem. Phys. **146**, 094104 (2017).
- ⁸³M. J. Harvey, G. Giupponi, and G. D. Fabritiis, J. Chem. Theory Comput. **5**, 1632 (2009).
- ⁸⁴Y. Zheng, B. Lindner, J.-H. Prinz, F. Noé, and J. C. Smith, J. Chem. Phys. **139**, 175102 (2013).
- ⁸⁵R. T. McGibbon and V. S. Pande, J. Chem. Phys. **142**, 124105 (2015).
- ⁸⁶J. D. Hunter, Computing In Science & Engineering **9**, 90 (2007).