Deep learning to decompose macromolecules into independent Markovian domains

Andreas Mardt[a,1], **Tim Hempel**[a,b,1], **Cecilia Clementi**[b,c], **and Frank Noé**[a,b,c,d,*]

[a]Freie Universität Berlin, Department of Mathematics and Computer Science, Germany; [b]Freie Universität Berlin, Department of Physics, Germany; [c]Rice University, Department of Chemistry, Houston, TX, USA; [d]Microsoft Research, Cambridge, UK

This manuscript was compiled on March 30, 2022

The increasing interest in modeling the dynamics of ever larger proteins has revealed a fundamental problem with models that describe the molecular system as being in a global configuration state. This notion limits our ability to gather sufficient statistics of state probabilities or state-to-state transitions because for large molecular systems the number of metastable states grows exponentially with size. In this manuscript, we approach this challenge by introducing a method that combines our recent progress on independent Markov decomposition (IMD) with VAMPnets, a deep learning approach to Markov modeling. We establish a training objective that quantifies how well a given decomposition of the molecular system into independent subdomains with Markovian dynamics approximates the overall dynamics. By constructing an end-to-end learning framework, the decomposition into such subdomains and their individual Markov state models are simultaneously learned, providing a data-efficient and easily interpretable summary of the complex system dynamics. While learning the dynamical coupling between Markovian subdomains is still an open issue, the present results are a significant step towards learning "Ising models" of large molecular complexes from simulation data.

VAMPnets | Markov state models | Machine Learning | Molecular Dynamics

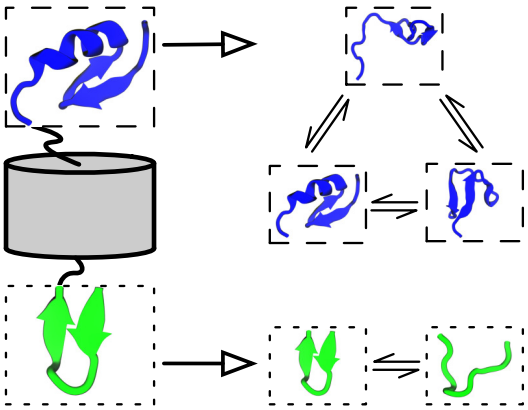

Fig. 1. The iVAMP concept as visualized by modeling dynamics of a protein that has two independent, flexible regions separated by a rigid barrel. iVAMPnets learn an assignment of the C- (blue/top) and N-termini (green/bottom) into independent subsystems from molecular dynamics trajectories (left column). Moreover, the dynamics of both termini are modeled with statistically independent VAMPnets (right column).

The understanding of protein function is often interlinked with understanding protein dynamics. Molecular dynamics (MD) simulations are a valuable tool to study these dynamics on an atomistic level (1–6). However, further methods are necessary to extract the statistically relevant information and to help overcome the discrepancy between feasible simulation length and the timescales of relevant processes. A common approach to enhance sampling of a specific process of interest is to bias the simulation along a reaction coordinate aligning with the process (7–13). In comparison, the Markov modeling approach (14–20) extracts kinetic information and tackles the sampling problem without requiring the definition of few predefined reaction coordinates by combining arbitrary numbers of short unbiased distributed simulations to model the long-timescale behavior of target systems. Consequently, multiple software packages (21, 22) have been developed over the last decade providing assistance in estimating these models. They often include a pipeline for feature selection (21–24), dimension reduction (25–31), clustering (32–35), transition matrix estimation (15, 19, 36, 37), and coarse graining (38–44). MSMs have been applied to a wide range of molecular biology problems such as protein aggregation (45–47) or ligand binding (48–50) and can be a valuable tool to understand experimental data on the atomistic scale (51, 52).

The necessity to assess a model's performance and thereby rank its quality encouraged the development of variational methods (53, 54), in particular the variational approach for Markov processes (VAMP) (55). This variational formulation has allowed us to replace the aforementioned pipeline with an end-to-end deep learning framework called VAMPnet (56), which simultaneously learns a dimension reduction of the molecular system to the collective variables best describing the rare event processes and a MSM on these variables. The framework can be used to further drive MD simulations along these learned collective variables (57, 58). We can also use this framework to estimate statistically reversible MSMs and incorporate constraints from experimental observables (59–61).

Despite these developments, there is a fundamental scaling problem in describing MD in terms of transitions between global system states: While the assignment of MD configurations to discrete global states representing the metastable groups of structures is an excellent model for small cooperative molecular systems, such as small to medium proteins, larger molecular systems (e.g. proteins with hundreds of amino acids) have an increasing number of subsystems whose dynamics are (nearly) independent (62) (Fig. 1). Consider, for example, a solution of N proteins which undergo transitions between their open and closed states independently when these proteins are

A.M., T.H. performed research (A.M. derived loss functions and implemented deep learning framework; T.H. designed method and developed test systems); A.M., T.H. analyzed data; A.M., T.H., C.C., F.N. designed research; A.M., T.H., C.C., F.N. wrote the paper.

The authors declare no conflicts of interests.

[1]A.M. and T.H. contributed equally to this work

*To whom correspondence should be addressed. E-mail: frank.noe@fu-berlin.de

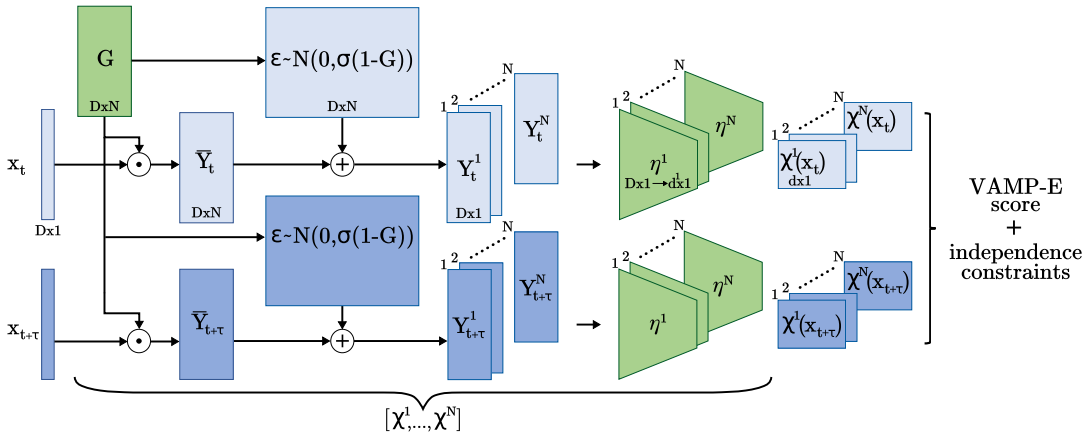

Fig. 2. Architecture of an iVAMPnet for N subsystems, where trainable parts are shaded green. Two lobes are given for configuration pairs $\mathbf{x}_t$ and $\mathbf{x}_{t+\tau}$, where the weights are shared. Firstly, the input features are element wise weighted $\bar{\mathbf{Y}}_t = \mathbf{G} \odot \mathbf{x}_t$ with a mask $\mathbf{G} \in \mathbb{R}^{D \times N}$, where each subsystem learns its individual weighting. The mask values can be interpreted as probabilities to which subsystem the input feature belongs. In order to prevent the subsequent neural network to reverse the effects of the mask, we draw for each input feature i and subsystem j an independent, normally distributed random variable $\epsilon_{ij} \sim \mathcal{N}(0, \sigma(1 - G_{ij}))$. This noise is added to the weighted features $\mathbf{Y}_t = \bar{\mathbf{Y}}_t + \boldsymbol{\epsilon}$. Thereby, the attention weight linearly interpolates between input feature and Gaussian noise, i.e., if the attention weight $G_{ij} = 1$, Y_{ij} carries exclusively the input feature x_i, if $G_{ij} = 0$, Y_{ij} is simple Gaussian noise. Afterwards, the transformed feature vector is split for each individual subsystem $\mathbf{Y}_t = [\mathbf{Y}_t^1, ..., \mathbf{Y}_t^N]$ and passed through the subsystem specific neural network $\boldsymbol{\eta}^i$. We call the whole transformation for a subsystem i the fuzzy state assignment $\boldsymbol{\chi}^i(\mathbf{x}_t) = \boldsymbol{\eta}^i(\mathbf{Y}_t^i)$.

dissociated and these transitions only (partially) couple when they are associated with other proteins. The number of global system states is 2^N, i.e. grows exponentially with the number of subsystems N (63, 64). This means any form of simulation or analysis which explicitly distinguishes global system states will not scale to large molecular systems.

At the same time, the (approximate) independence between subsystems is also key to the solution of the problem. A scalable solution needs to address two separate issues: (a) dividing the protein system into approximately Markovian subsystems and (b) learning the coupling between them. Olsson & Noé (63) made a first attempt at (b), by learning a dynamic graphical model between predefined subsystems. This approach leads to a graphical model, or Markov random field, resembling Ising or Potts models in physics, with the key difference that both the definition of the individual subsystems or "spins" as well as their transition dynamics need to be learned. In contrast, Hempel et al. (64) proposed a solution for (a) by approximating the global system dynamics as a set of independent (uncoupled) Markov models (termed Independent Markov decomposition, IMD). They furthermore propose a pairwise independence score of features, which allows to detect nearly uncoupled regions where independent Markov state models can be estimated subsequently.

In this manuscript, we present a joint IMD and VAMP approach (termed independent VAMPnet, or shorthand iVAMPnet) that significantly advances our ability to identify approximately independent Markovian subsystems (issue a) by generalizing IMD to neural network basis functions. iVAMPnets are an integrated end-to-end learning approach that decomposes the macromolecular structure into subsystems that are dynamically weakly coupled, and estimates a VAMPnet for each of these subsystems to promote a comprehensible analysis of the subsystem dynamics (Fig. 1). In comparison to previous implementations of IMD, our approach learns an optimal decomposition into independent subsystems and can find collective variables that are nonlinear combinations of the input features.

Theory

Markov state models and Koopman models. Markovian dynamics can be modeled by the transition density:

$$p_\tau(\mathbf{y}|\mathbf{x}) = \mathbb{P}(\mathbf{x}_{t+\tau} = \mathbf{y}|\mathbf{x}_t = \mathbf{x}),$$

which is the probability density to observe configuration $\mathbf{y}$ at time $t + \tau$ given that the system was in configuration $\mathbf{x}$ at time t. Based on the transition density we can characterize the time evolution of a probability density χ as:

$$\chi_{t+\tau}(\mathbf{y}) = \int p_\tau(\mathbf{y}|\mathbf{x})\chi_t(\mathbf{x})\mathrm{d}\mathbf{x}. \qquad [1]$$

By discretizing the molecular state space in a suitable way and defining a transition matrix $\mathbf{T}$ between discrete states, we can linearize this equation as:

$$\boldsymbol{\chi}_{t+\tau}(\mathbf{y}) = \mathbf{T}_\tau^T \boldsymbol{\chi}_t(\mathbf{x}) \qquad [2]$$

This is the equation of a Markov state model, where the element i of the vector $\boldsymbol{\chi}_{t+\tau}(\mathbf{y})$ is the probability to be in the discrete state i at time $t + \tau$. Furthermore, the transition matrix elements $(\mathbf{T}_\tau)_{ij}$ describe the transition probabilities for jumping to state j given state i within a time τ. In the case of fuzzy state assignments, e.g., as with VAMPnets, Eq. (2) describes the more general Koopman model (65) and $\mathbf{T}_\tau$ becomes the Koopman matrix. This means that probability densities are still propagated but the matrix elements cannot be interpreted as transition probabilities.

We now seek to find a state assignment $\boldsymbol{\chi}$ and model matrix $\mathbf{T}$ that satisfy Eq. (2) and also succeed in predicting the long-time behavior, i.e., for multiples of the lag time τ. VAMP (55) tells us that an optimal solution is reached when $\boldsymbol{\chi}$ can span the left $(\psi_1, ..., \psi_k)^T$ and right singular functions $(\phi_i, ..., \phi_k)^T$ of the transition operator. They can be found by maximizing the singular values of a matrix that can be estimated from simulation data (see Eqs. 3-7 in Methods). In the case of a VAMPnet (56), deep neural networks are trained by maximizing the VAMP score, so as to represent optimal fuzzy state assignments.

In equilibrium, the singular functions correspond to the eigenfunctions of the Markov state model and the singular values to its eigenvalues. As the Koopman model still propagates densities, it is instructive to inspect the eigenfunctions and implied timescales of $\mathbf{T}$ since they describe the slow dynamics of a given system.

iVAMPnets and iVAMP-score. To implement iVAMPnets, we need to bridge the gap between the deep neural networks of VAMPnets and the spatial decomposition of independent Markov models. The general idea is to set up multiple parallel VAMPnets, each modeling the Markovian dynamics of a separate, independent subsystem of the molecule, together with an attention mechanism that identifies these subsystems. Thus, each independent VAMPnet should only receive the time dependent molecular geometry features representing its specific subsystem. For example, such an attention mechanism could separate different protein domains and channel the data of individual domains to separate VAMPnets. We therefore develop an architecture that combines a meaningful attention mechanism and parallel VAMPnets and trains them with a loss function that simultaneously promotes dynamic independence between the subsystems and slow kinetics within each subsystem (Fig. 2). iVAMPnets are designed to optimize both these objectives simultaneously.

In practice, we extract all time-lagged data pairs $\mathbf{x}_t, \mathbf{x}_{t+\tau}$ that contain all molecular geometry features (e.g., distances, contacts, torsions) of our simulation data and pass them through the architecture presented in Fig. 2. The data is fed through an attention mechanism (represented by the matrix $\mathbf{G}$) that yields subsystem specific vectors $\mathbf{Y}_t^i$, each of which attends to features relevant for subsystem i. These vectors then serve as inputs to N parallel feature transformations $\boldsymbol{\eta}^i$ (parallel VAMPnets) which transform those into output features $\boldsymbol{\chi}^1, \dots \boldsymbol{\chi}^N$ (with $\boldsymbol{\chi}^i(\mathbf{x}_t) = \boldsymbol{\eta}^i(\mathbf{Y}_t^i(\mathbf{x}_t))$) that represent slow collective coordinates or directly fuzzy assignments to metastable Markov states of each molecular subsystem. Equipped with the state assignments, we can compute correlation matrices (Eq. 3) and derive a Koopman model matrix from those (Eq. 4). As in VAMPnets, the feature transformations $\boldsymbol{\eta}^1, \dots \boldsymbol{\eta}^N$ are represented by deep neural networks. In the present study we use multilayer perceptrons with a SoftMax output layer representing fuzzy state assignments. However, other architectures could be chosen, e.g. graph convolution networks when parameter sharing is desired (66, 67), and a linear output layer could be chosen if the aim is to represent slow collective variable rather than discrete states (57, 58). The parameters of the feature transformations $\boldsymbol{\eta}$ and the attention matrix are learned end-to-end via backpropagation.

To train the model we develop a loss function that (i) maximizes the VAMP score of each VAMPnet, assuming that they describe independent dynamics (Methods, Eqs. 11-14), and (ii) minimizes a term that penalizes statistical dependence between these subsystems (Methods, Eqs. 15-17). The final loss function (Eq. 18) is a weighted sum of both terms. The remaining constraints (Eq. 16) can be used as post training validation metrics for testing to which degree dynamically independent subsystems were found.

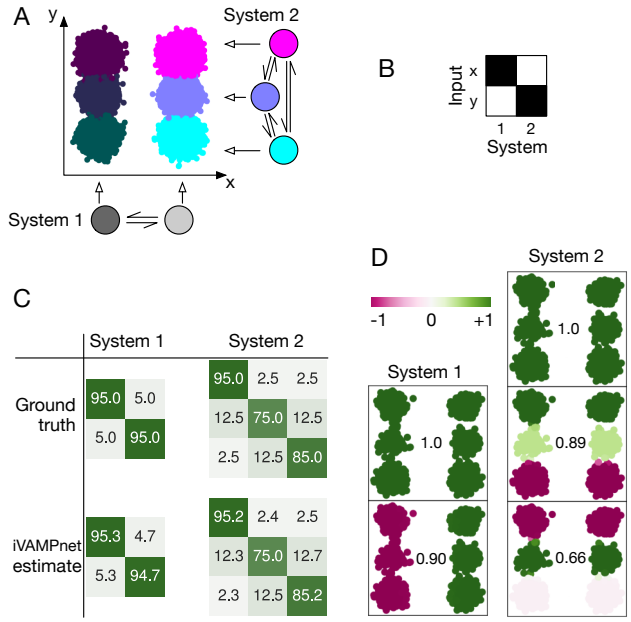

Fig. 3. Hidden Markov state model as a benchmark example for independent subsystems: **a)** 2 subsystems with 2 and 3 states emit independently to an x and y axis, respectively. The corresponding 2D space embeds all 6 global states. **b)** The learned mask shows that each subsystem focuses on one input dimension. **c)** The estimated subsystem transition matrices are compared with the ground truth. **d)** Subsystem eigenfunctions and corresponding eigenvalues as found by iVAMPnet. Independent processes are recovered from the 2D data.

Results

The iVAMPnet architecture, which is implemented using *PyTorch* (68), is depicted in Fig. 2. Generally, various neural network architectures are possible; we here choose fully connected feed forward neural networks with up to 5 hidden layers with 100 nodes each. The scripts to reproduce the results including the details for the training routine, choice of hyper-parameters, and network architecture can be found in our GitHub repository. We note that an implementation of VAMPnets is available in the current version of DeepTime (69).

Benchmark model with 2 independent subsystems. We first demonstrate that iVAMPnets are capable of decomposing a dynamical system into its independent Markovian subsystems based on observed trajectory data using an exactly decomposable benchmark model (Fig. 3).

Akin to the protein illustrated in Fig. 1, we define a system that consists of two independent subsystems with two and three states, respectively. It is modeled by two transition matrices with the corresponding number of states. We sample a discrete trajectory with each matrix (100k steps) (69). The global state is defined as a combination of these discrete states. The discrete subsystem states are now interpreted as the hidden states of hidden Markov models (70) that emit to separate, subsystem-specific dimensions of a 2D space. The output of each subsystem is modeled with Gaussian noise $N(\mu_i, \tilde{\sigma}) \in \mathbb{R}$ that is specific to the state that the system is in, specified by the mean μ_i, and a constant $\tilde{\sigma}$. The two state subsystem therefore describes a jump process between Gaussian basins along the x-axis and the three state along the y-axis, respectively (Fig. 3a). These variables compare to collective variables of the green (x) and blue (y) system depicted in Fig. 1. Please

note that while in this benchmark system the relevant slow collective variables are known, iVAMPnets are generally capable of finding them (cf. 10D hypercube benchmark model and Synaptotagmin-C2A).

Since the generative benchmark model consists of perfectly independent subsystems and the pair already describes the global system, our method can simply be optimized for the global VAMP-E score (Eq. 13) without the need for any further constraints. We train a model with a two and three state subsystem at a lag time of $\tau = 1$ step.

Once trained, the iVAMPnet yields a model of the dynamics in each of the identified subsystems. As expected, we find that the estimated transition matrices for both subsystems closely agree with the ground truth (Fig. 3c). To additionally assess the slow subsystem dynamics in more detail, we borrow concepts from MSM analysis and conduct an eigenvalue decomposition of the iVAMPnet models (cf. VAMPnets). The analysis of the eigenfunctions demonstrates that, by construction, the system exhibits one independent process along the x-axis ($\lambda_1 = 0.90$) and two along the y-axis ($\lambda_2 = 0.89$ and $\lambda_4 = 0.66$) (Fig. 3d). In contrast, we note that in the picture of *global* states, two additional processes would appear as a result of mixing the independent processes (cf. SI Appendix, Fig. S1), which makes the combined dynamical model more challenging to analyze, whereas the iVAMPnet analysis remains straightforward and simple.

Besides the dynamical models, our iVAMPnet yields assignments between input features and subsystems. We find that the method correctly identifies the two state system as the x-axis and the three states as the y-axis feature, respectively (Fig. 3b).

10D hypercube benchmark model. In a next step we test the iVAMPnet approach with ten 2-state subsystems, which corresponds to 1024 global states (Fig. 4a,b). As before, the dynamics is generated by ten independent Hidden Markov state models with unique timescales. The system is split into five pairs of subsystems, and the two coordinates governing the transition dynamics of each pair are rotated in order to make them more difficult to separate (Fig. 4a). Additionally, we make the learning problem harder by adding ten noise dimensions such that the global system lives on a 10-dimensional hypercube embedded in a 20 dimensional space.

Although the subsystems are perfectly independent, we will estimate an iVAMPnet with the VAMP-E score in a pairwise fashion, thereby avoiding to estimate expensively large correlation matrices in $\mathbb{R}^{1024\times1024}$. As this is only justified if all systems are independent, we additionally enforce Eq. (14) during training by minimizing Eq. (18) and thereby rule out that any two subsystems approximate the same process.

The iVAMPnet estimation yields subsystem models which, as common in MSM analysis, can be validated by testing whether their implied relaxation timescales are converged in the model lag time τ. We find that the implied timescales learned by the iVAMPnet are indeed converged and accurately reproduce the ground truth (Fig. 4d). We note that in addition to the timescales of the individual subsystems that are identified by the iVAMPnet, a global model would also contain all timescales that result from products of eigenvalues, resulting in a total of 1024 timescales. Thus, the iVAMPnet analysis provides a much simpler and more concise model than a global MSM or VAMPnet would.

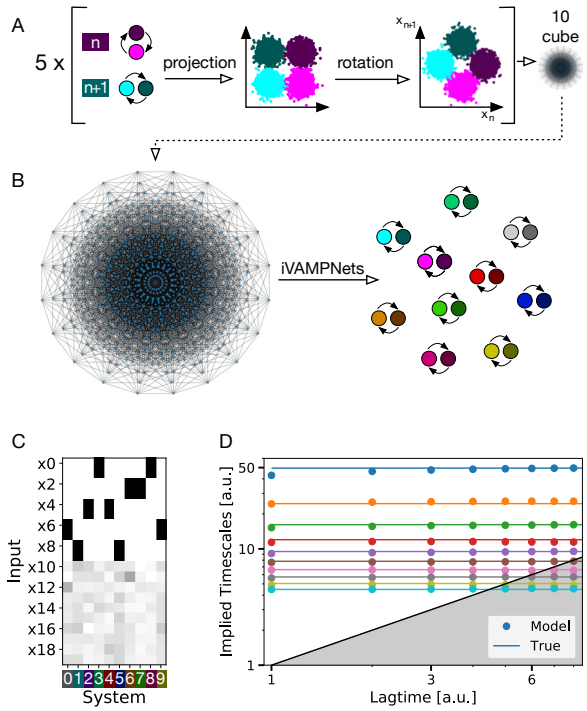

Fig. 4. Hidden Markov State model with 1024 global states forming a 10D hypercube embedded in a 20D space. **a)** The hypercube is composed of ten independent 2 state subsystems. A pair of 2 subsystems always lives in a common rotated 2D-manifold. Therefore, two subsystems need the same input features to be well approximated. **b)** 2D depiction of the hypercube in an orthographic projection (71, 72), where the global system can jump freely between all 1024 vertices, and the ten 2-state models retrieved from it by the iVAMPnet. **c)** Learned mask shows that for each subsystem, the network assigns 2 highly important input features which are shared with exactly one other subsystem, mirroring the rotated input space. Noise dimensions (x10-x19) are assigned low importance values. **d)** Implied timescales of all 10 subsystems learned by our method (dots) approximate the underlying true timescales (lines).

Furthermore, the subsystem assignment mask indicates that the method correctly assigns high importance weight to two input features for each model (Fig. 4c). Therefore, the method proves its capability of decomposing a noisy, high dimensional global system into its independent sub-processes in a data efficient way.

Synaptotagmin-C2A. Finally, we test iVAMPnets on an all-atom protein system. In comparison to our benchmark examples, we expect the underlying global dynamics to be only approximately decomposable into independent subsystems. Our test system is the C2A domain of synaptotagmin that was described previously (73); it plays a crucial role in the regulation of neurotransmitter release (74). It was shown to consist of approximately uncoupled subsystems containing the calcium binding region (CBR) and the C78 loop, respectively (64).

First, we attempted to model the protein with a global model, i.e., with a single (regular) VAMPnet. Indeed, this approach failed because there were not enough simulation statistics to estimate a reversibly connected transition model between all global metastable states, resulting in diverging implied timescales (SI Appendix, Global model of synaptotagmin and Fig. S2). This is exactly the scenario where iVAMPnets should provide an advantage, by only relying on locally rather than globally converged transition statistics.

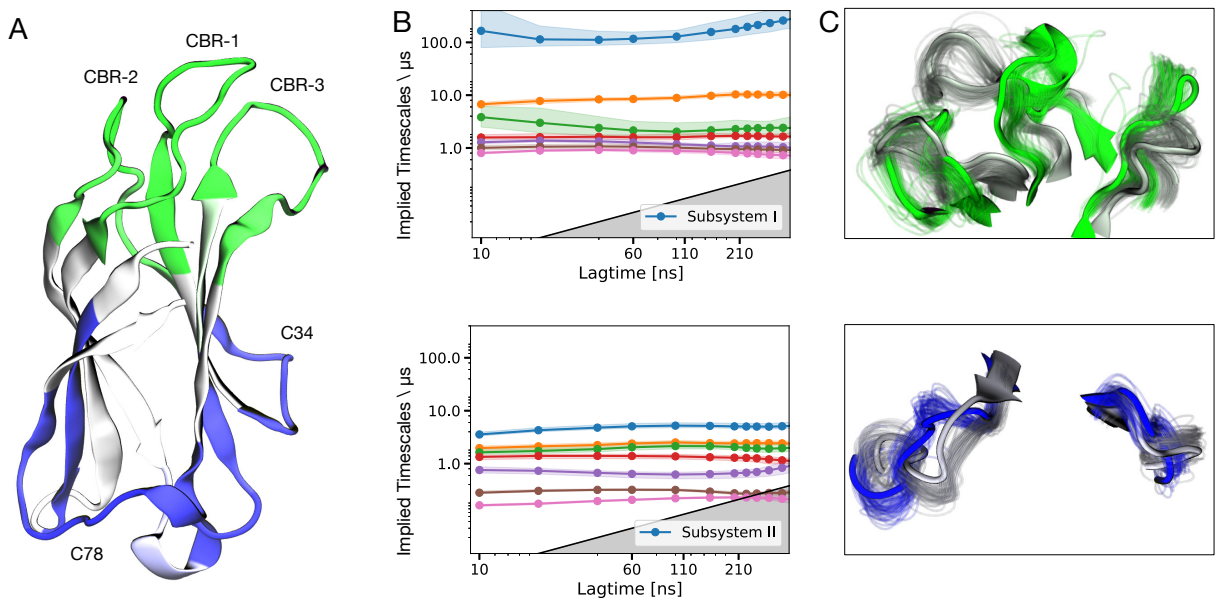

Fig. 5. iVAMPnet of synaptotagmin-C2A with 2 subsystems and 8 states each. **a)** Importance values of the trainable mask depicted as color-coded protein secondary structure, indicating assignment to subsystem I (II) in green (blue). **b)** Implied timescales of the 2 subsystems with a 90% percentile over 20 runs. **c)** Superposed representative structures of both extrema of the slowest resolved eigenfunctions of each subsystem (residues not assigned a high importance value or not showing significant movement are omitted for clarity). The slowest process of subsystem 1 changes between green and gray structures showing an orchestrated movement of the full Calcium Binding Region (CBR1, CBR2, and CBR3). The slowest process of the second subsystem occurs between the blue and gray structures and describes a combined movement of C78 and C34.

Next, we train an iVAMPnet to seek two subsystems of 8 states each at a lag time of $\tau = 10$ ns where we enforce constraint (14) to find uncoupled subsystems.

The trained iVAMPnet identifies one subsystem comprising all three CBR loops (CBR-1, CBR-2, CBR-3; Fig. 5a). The second subsystem consists not only of the aforementioned C78 loop but also of the loop connecting beta sheets 3 and 4 (75) (termed C34 henceforth). When mapping the residue positions on the protein structure it becomes obvious that the two subsystems are physically well separated (Fig. 5a), supporting the conclusion that both regions are only weakly coupled (64).

The implied timescales of both systems are approximately constant in the model lag time τ. Most timescales are in the range of $1 - 10$ μs, with the exception of one much slower process with a 100 μs relaxation time found in the first subsystem (Fig. 5b), which has not been found previously. Analysis of the structural changes governing this process reveals that it involves an orchestrated transition of all CBR loops (Fig. 5c). Such a process could however not be resolved by the previous study (73) where the CBR was modeled as individual loops. The process of the second system involves a simultaneous movement of the C78 and C34 loops (Fig. 5c). Please compare SI Appendix, Fig. S3 for further validation.

Although estimating a global VAMPnet model for synaptotagmin was not feasible given the sparse data sample, iVAMPnets use the same data efficiently and estimate a statistically valid dynamical model. This result is especially striking because the iVAMPnet approach also simplifies the subsequent task of interpreting models by separating dynamically independent protein domains.

Counterexample: folding of the villin miniprotein. Finally, we conducted an experiment on a villin protein folding trajectory (76) as a negative example. Small proteins such as villin are typically cooperative, i.e., the slowest processes related to folding involve all residues (SI Appendix, Villin: A non-decomposable system). Thus, these processes cannot be resolved when decomposing the system into several subsystems. Indeed, we find that a splitting into two subsystems with two states each results in timescales that are not converged, and whose relaxation processes approximate a partial folding on disjoint areas (cf. SI Appendix, Fig. S4).

Testing statistical independence of the learned dynamical subsystems. As constraint Eq. (14) was used as a penalty during training (as independence score Eq. (17)), we assess the validity of an estimated subsystem assignment by evaluating the constraints that were not enforced during training (Eq. 15) as post-training independence scores M_U, M_V, and M_{UV} (defined in Eq. (16)). Low values for M_U and M_V imply that the constructed left and right singular functions are indeed valid candidates for singular functions in the global state space. A small value for M_{UV} indicates that the kinetics in the global state space is well predicted by the Kronecker product of subsystem models. We find that the three metrics are well suited to indicate independence of the learned subsystems (Tab. 1). Out of the tested systems only villin cannot be split into independent parts (all scores > 0.1). In comparison, the benchmark models and synaptotagmin can be decomposed into statistically uncoupled subsystems (all scores < 0.01). The slightly increased values for synaptotagmin suggest that its subsystems might be weakly coupled.

Discussion

We have proposed an unsupervised deep learning framework that, using only molecular dynamics simulation data, learns to decompose a complex molecular system into subsystems

	M_U	M_V	M_{UV}	M_R
Benchmark 2	0.0058	0.0059	0.0055	0.0002
10-Cube	0.0039	0.0039	0.0046	0.0005
Synaptotagmin	0.0068	0.0068	0.0071	0.0016
Villin	0.1353	0.1364	0.1493	0.0021

Table 1. Post-training independence validation. The scores in columns 1-3 (M_U, M_V, M_{UV}, cf. Eq. (16)) are computed from independence constraints that were not enforced during the training. The score in the last column (M_R) is used during the training and shown for reference. The three post-training validation scores M_U, M_V, and M_{UV} indicate that the final subsystems of both benchmark examples and synaptotagmin are indeed independent, whereas the scores for villin strictly oppose this conjunction. The standard deviations (SD) over 10 different runs are on the order of 10^{-5} for all systems except villin, which has an SD $\sim 10^{-4}$.

which behave as approximately independent Markov models. Thereby, iVAMPnet is an end-to-end learning framework that points a way out of the exponentially growing demand for simulation data that is required to sample increasingly large biomolecular complexes.

Specifically, we have developed and demonstrated iVAMPnets for molecular dynamics, but the approach is, in principle, also applicable to different application areas, such as fluid dynamics. The specific implementation, such as the representation of the input vectors $\mathbf{x}_t$ and the neural network architecture of the $\boldsymbol{\chi}$-functions, depend on the application and can be adapted as needed.

We now have a hierarchy of increasingly powerful models ranging from MSMs over VAMPnets to iVAMPnets. MSMs always consist of (1) a state space decomposition and (2) a Markovian transition matrix governing the dynamics between these states. VAMPnets provide a deep learning framework for MSMs, and thereby (3) learn the collective coordinates in which the state space discretization (1) is best made. iVAMPnets additionally learn (4) a physical separation of the molecular system into subsystems, each of which has its own slow coordinates, Markov states, and transition matrix.

We have demonstrated that iVAMPNets are a powerful multiscale learning method that succeeds in finding and modeling molecular subsystems when these subsystems indeed evolve statistically independently. Additionally, iVAMPnets are capable of learning from high dimensional MD data. To prove that point, we have demonstrated that the synaptotagmin C2A domain is decomposable into two almost independent Markov state models. Importantly, we have shown that this dynamical decomposition of synaptotagmin C2A succeeds while an attempt to model the system with a global Markov state model fails due to poor sampling. This is a direct demonstration that iVAMPnets are statistically more efficient than VAMPnets, MSMs or other global-state models and may indeed scale to much larger systems.

We note, however, that iVAMPnets do not learn how the subsystems are coupled, and are therefore, in their current form, only applicable to molecular systems that consist of uncoupled or weakly coupled subsystems. Following up on Ref. (63) and introducing coupling parameters that describe how the learned MSMs are coupled, is subject to ongoing research.

Besides the usual hyperparameter choices in deep learning approaches, iVAMPnets require the specification of the number of sought subsystems. This choice can be guided by training an iVAMPnet for different numbers of subsystems and then interrogating the independence scores (Eq. 17 and Eqs. 16) to choose a decomposition where statistical independence is optimal. Furthermore, this choice can be guided by the number of structural domains or by using the network-based approach presented in Ref. (64).

iVAMPnets can be improved and further developed in multiple ways, e.g. by employing more advanced network architectures, e.g. graph neural networks, where parameters could be shared across subsystems. This might result in higher quality models and a greater robustness against the hyperparameter choice.

In summary, iVAMPnets pave a possible path for modeling the kinetics of large biological systems in a data-efficient and interpretable manner.

Materials and Methods

VAMPnets. Since an iVAMPnet implements multiple parallel VAMPnets representing the kinetics of separate independent subsystems, we will introduce VAMPnets first. Given the feature functions $\boldsymbol{\chi}$ (we omit the upper subsystem index i for the sake of clearness here), we can compute the following correlation matrices:

$$\begin{aligned} \mathbf{C}_{00} &= \frac{1}{L}\sum_t \boldsymbol{\chi}(\mathbf{x}_t)\boldsymbol{\chi}(\mathbf{x}_t)^T \\ \mathbf{C}_{0\tau} &= \frac{1}{L}\sum_t \boldsymbol{\chi}(\mathbf{x}_t)\boldsymbol{\chi}(\mathbf{x}_{t+\tau})^T \\ \mathbf{C}_{\tau\tau} &= \frac{1}{L}\sum_t \boldsymbol{\chi}(\mathbf{x}_{t+\tau})\boldsymbol{\chi}(\mathbf{x}_{t+\tau})^T, \end{aligned} \quad [3]$$

where L is the number of collected data pairs in the simulations. The approximation of the singular functions and values can be estimated via the singular value decomposition (SVD) of the following matrix $\bar{\mathbf{K}}$:

$$\bar{\mathbf{K}} = \mathbf{C}_{00}^{-1/2}\mathbf{C}_{0\tau}\mathbf{C}_{\tau\tau}^{-1/2} = \mathbf{A}\mathbf{K}\mathbf{B}^T \quad [4]$$

$\mathbf{K}$ is the diagonal matrix of approximated singular values corresponding to the left and right singular functions:

$$\mathbf{f}^T(\mathbf{x}_t) = \boldsymbol{\chi}(\mathbf{x}_t)^T\mathbf{U} = \boldsymbol{\chi}(\mathbf{x}_t)^T\mathbf{C}_{00}^{-1/2}\mathbf{A} \quad [5]$$

$$\mathbf{g}^T(\mathbf{x}_{t+\tau}) = \boldsymbol{\chi}(\mathbf{x}_{t+\tau})^T\mathbf{V} = \boldsymbol{\chi}(\mathbf{x}_{t+\tau})^T\mathbf{C}_{\tau\tau}^{-1/2}\mathbf{B}. \quad [6]$$

The matrices $\mathbf{U}$ and $\mathbf{V}$ construct the left and right singular functions from the individual state assignments. The optimal state assignments can be found by maximizing the VAMP-E score:

$$\mathcal{R}_E = \mathrm{tr}[2\mathbf{K}\mathbf{U}^T\mathbf{C}_{0\tau}\mathbf{V} - \mathbf{K}\mathbf{U}^T\mathbf{C}_{00}\mathbf{U}\mathbf{K}\mathbf{V}^T\mathbf{C}_{\tau\tau}\mathbf{V}]. \quad [7]$$

Given trained state assignments $\boldsymbol{\chi}(\mathbf{x}_t)$ and correlation matrices Eq. (3), the Koopman matrix $\mathbf{T}$ can then be evaluated as:

$$\mathbf{T} = \mathbf{C}_{00}^{-1}\mathbf{C}_{0\tau}. \quad [8]$$

Further, we can estimate the eigenfunction $\boldsymbol{\varphi}$ and timescales t_i by its eigendecomposition $\mathbf{T} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1}$:

$$\boldsymbol{\varphi}(\mathbf{x}) = \mathbf{Q}^T\boldsymbol{\chi}(\mathbf{x}), \quad [9]$$

$$t_i = \frac{-\tau}{\log(\Lambda_{ii})}. \quad [10]$$

Please note that this operation is only possible if the eigendecomposition is (approximately) real-valued, a condition that is met for the presented application cases.

iVAMPnets

Given N individual subsystem models, the global system state can be given by the Kronecker product of all subsystem states:

$$\boldsymbol{\chi}^G(\mathbf{x}_t) = \bigotimes_i \boldsymbol{\chi}^i(\mathbf{x}_t) \quad [11]$$

and by computing the global correlation matrices $(\mathbf{C}_{00}^G, \mathbf{C}_{0\tau}^G, \mathbf{C}_{\tau\tau}^G)$ from Eqs. (3) using $\boldsymbol{\chi}^G$. We note that this step does not require that we have independent Markovian models, but it is simply a formalism to express global states in terms of a combination of local states.

Furthermore, we can construct a candidate for the global Koopman model from the subsystem models by combining the individual singular values and vectors with a Kronecker product (64):

$$\hat{\mathbf{K}}^G = \bigotimes_i \mathbf{K}^i \qquad \hat{\mathbf{U}}^G = \bigotimes_i \mathbf{U}^i \qquad \hat{\mathbf{V}}^G = \bigotimes_i \mathbf{V}^i. \quad [12]$$

The matrices $\hat{\mathbf{U}}^G$ and $\hat{\mathbf{V}}^G$ map the global state assignments onto the constructed singular functions and are computed from the local matrices as defined in Eqs. (5-6). The diagonal matrix $\hat{\mathbf{K}}^G$ encodes the singular values and is computed from the subsystem singular value matrices via Eq. (4).

In order to evaluate the performance of the constructed model to predict the dynamics in the global state space, the VAMP-E validation (55) score can be exploited. The VAMP-E score measures the difference between the estimated Koopman model and the true dynamics.

$$\begin{aligned} \mathcal{R}_E^G &= \mathrm{tr}[2\hat{\mathbf{K}}^G\hat{\mathbf{C}}_{0\tau}^G - \hat{\mathbf{K}}^G\hat{\mathbf{C}}_{00}^G\hat{\mathbf{K}}^G\hat{\mathbf{C}}_{\tau\tau}^G] \\ \hat{\mathbf{C}}_{00}^G &= (\hat{\mathbf{U}}^G)^T\mathbf{C}_{00}^G\hat{\mathbf{U}}^G \\ \hat{\mathbf{C}}_{0\tau}^G &= (\hat{\mathbf{U}}^G)^T\mathbf{C}_{0\tau}^G\hat{\mathbf{V}}^G \\ \hat{\mathbf{C}}_{\tau\tau}^G &= (\hat{\mathbf{V}}^G)^T\mathbf{C}_{\tau\tau}^G\hat{\mathbf{V}}^G. \end{aligned} \quad [13]$$

Here, the VAMP-E score is evaluated for the global state assignments $\bigotimes_i \boldsymbol{\chi}^i$ (as encoded in $\mathbf{C}_{00}^G, \mathbf{C}_{0\tau}^G, \mathbf{C}_{\tau\tau}^G$) mapped on the constructed singular functions (as encoded in $\hat{\mathbf{U}}^G, \hat{\mathbf{V}}^G$). If the subsystems are independent the constructed singular functions are optimal and the singular values of the global system are indeed the product of the singular values of the subsystems (see SI Appendix, Independent Koopman operators). Therefore, for Markov independent systems it holds that

$$\mathcal{R}_E^G = \prod_i \mathcal{R}_E^i. \quad [14]$$

It is a necessary condition to be fulfilled for independent subsystems which can be extended by Eq. (15). We note that Eq. (14) has a similar appearance as the *dependency* score in previous work (64), however here the validation score is deployed for $\mathcal{R}_E^G$ which includes an additional mapping on the constructed model instead of simply estimating the VAMP score of the global state vectors.

Given N subsystem assignments $\boldsymbol{\chi}^i$, we are in principle equipped to train their parameters for the VAMP-E score and to compute a global model from an all-subsystem Kronecker product.

Conditions for independent systems. For Markov independent systems, the singular values and functions that are constructed by the Kronecker product match the true global ones. In addition to Eq. (14), it therefore holds:

$$\begin{aligned} (\hat{\mathbf{U}}^G)^T\mathbf{C}_{00}^G\hat{\mathbf{U}}^G &= \mathbf{1} \\ (\hat{\mathbf{V}}^G)^T\mathbf{C}_{\tau\tau}^G\hat{\mathbf{V}}^G &= \mathbf{1} \\ (\hat{\mathbf{U}}^G)^T\mathbf{C}_{0\tau}^G\hat{\mathbf{V}}^G &= \hat{\mathbf{K}}^G, \end{aligned} \quad [15]$$

where the first two equations guarantee the orthonormality of the constructed singular functions. The latter verifies that the left and right singular functions correlate as predicted by the Kronecker product of the singular values.
These conditions can be translated to the following scores to quantify the fulfillment of them as a post-training validation:

$$\begin{aligned} M_U &= |(\hat{\mathbf{U}}^G)^T\mathbf{C}_{00}^G\hat{\mathbf{U}}^G - \mathbf{1}| \\ M_V &= |(\hat{\mathbf{V}}^G)^T\mathbf{C}_{\tau\tau}^G\hat{\mathbf{V}}^G - \mathbf{1}| \\ M_{UV} &= |(\hat{\mathbf{U}}^G)^T\mathbf{C}_{0\tau}^G\hat{\mathbf{V}}^G - \hat{\mathbf{K}}^G| \end{aligned} \quad [16]$$

$$M_R = \frac{|\mathcal{R}_E^G - \prod_i \mathcal{R}_E^i|}{\mathcal{R}_E^G} \quad [17]$$

The norms denote simple means. The last score, M_R, is enforced during training in a pairwise fashion (cf. Eq. 18).

Loss function. For training neural networks, a loss function L must be defined, which will be minimized during training. Since we aim to maximize the VAMP-E validation score Eq. (13), we construct the loss by its negative value. As a consequence of the conditions for independence Eq. (14) has to hold, which we add to the loss function weighted by ξ. Finally, we evaluate the scores only pairwise, to escape the growth of the global state space, and sum over all possible pairs i, j, resulting in:

$$L = -\sum_{i<j} \mathcal{R}_E^{ij} + \xi \sum_{i<j} \frac{||\mathcal{R}_E^{ij} - \mathcal{R}_E^i\mathcal{R}_E^j||}{\mathcal{R}_E^{ij}}. \quad [18]$$

Here, $\mathcal{R}_E^{ij}$ measures the quality of the constructed Koopman model of subsystems i and j and is computed using Eq. (13).

Network architecture. Given a global system, which we want to decompose into N subsystems, and a time series of input features $\{\mathbf{x}_t\}_{t=1,\dots T}$, $\mathbf{x}_t \in \mathbb{R}^{D\times 1}$, we pass the features through a mask $\mathbf{G} \in \mathbb{R}^{D\times N}$, which weights each input differently for each subsystem, before the result are transformed individually by the N independent state assignment functions $\boldsymbol{\eta}^i$. It should be mentioned that the mask is merely introduced for interpretability reasons and is not essential to find independent subsystems. If the mask was omitted, the extraction of the relevant features would simply be transferred to the downstream neural networks, remaining hidden to the practitioner.

The weighted input is assessed by an element wise multiplication $\bar{\mathbf{Y}}_t = \mathbf{G} \odot \mathbf{x}_t$. In order to prevent the neural networks to reverse the weighting of the mask in its consecutively layers, we draw for each input feature i and subsystem j an independent, normally distributed random variable $\epsilon_{ij} \sim \mathcal{N}(0, \sigma(1 - G_{ij}))$. This noise is added to the weighted features:

$$\mathbf{Y}_t = \bar{\mathbf{Y}}_t + \boldsymbol{\epsilon}. \quad [19]$$

Thereby, the attention weight linearly interpolates between input feature and Gaussian noise, i.e., if the attention weight $G_{ij} = 1$, Y_{ij} carries exclusively the input feature x_i, if $G_{ij} = 0$, Y_{ij} is simple Gaussian noise. By tuning the noise scaling σ a harder assignment by $\mathbf{G}$ can be enforced. Afterwards, the transformed feature vector is split for each individual subsystem $\mathbf{Y}_t = [\mathbf{Y}_t^1, ..., \mathbf{Y}_t^N]$ and passed through the subsystem specific neural network $\boldsymbol{\eta}^i$ resulting in feature transformations $\boldsymbol{\chi}^i(\mathbf{x}_t) = \boldsymbol{\eta}^i(\mathbf{Y}_t^i)$. These features are then used to estimate the Koopman models.

Constructing the mask. To train an interpretable mask, we use the following three premises:

1. A single subsystem should not focus on all input features.
2. Different subsystems compete for high weights for the same feature.
3. All weights should be in the range $[0, 1]$ and the matrix should be sparse.

Therefore, the mask is constructed by trainable weights $\mathbf{g} \in \mathbb{R}^{D\times N}$ which are first processed by a softmax function which normalizes along the input feature axis $\mathbf{g}_1 = \mathrm{softmax}(\mathbf{g}, \mathrm{dim}{=}0)$. Thereby, if a subsystem focuses on one part of the features, a lower weight for the other parts is expected following the first premise.

In a next step, weights which are lower than a threshold θ are clipped to zero $\mathbf{g}_2 = \mathrm{relu}(\mathbf{g}_1 - \theta)$ to guarantee sparsity. Since input features could be negligible for all subsystems, a dummy system is added which has a constant value $\mathbf{c} \in \mathbb{R}^{D\times 1}$ for all features $\mathbf{g}_3 = [\mathbf{g}_2, \mathbf{c}]$. Consequently, the weights of all subsystems and the dummy

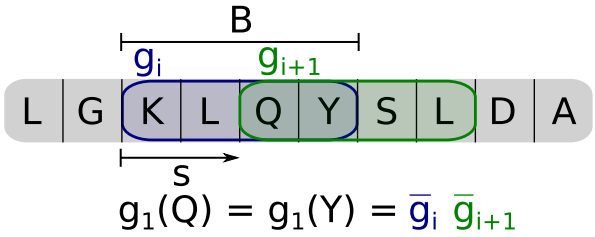

Fig. 6. Attention scheme for amino acid chain. Windows of size B are placed along the chain with a step size of s resulting into W many windows. A trainable weight $\mathbf{g} \in \mathbb{R}^{W \times N}$ is assigned for a window in each subsystem which are made positive and normalized along the window axis through a softmax $\bar{\mathbf{g}} = \text{softmax}(\mathbf{g}, \text{dim=0})$. Here a window size of $B = 4$ and a step size of $s = 2$ is chosen. As a consequence the weight of the amino acid glutamine (Q) is given as the product of the two windows it is part of $\mathbf{g}_1(Q) = \bar{\mathbf{g}}_i \bar{\mathbf{g}}_{i+1}$. The choice of the step size determines how many neighboring amino acids have the exact same weight within a subsystem, which applies here for the tyrosine (Y). Together with the window size it is regulated how many residues share parts of their weights. Hence, the serine (S) shares the weight $\bar{\mathbf{g}}_{i+1}$ with the previous two amino acids $\mathbf{g}_1(S) = \bar{\mathbf{g}}_{i+1}\bar{\mathbf{g}}_{i+2}$, which has a smoothing effect on the attention mechanism along the chain.

system are normed for each feature $\mathbf{g}_4 = \mathbf{g}_3/\text{sum}(\mathbf{g}_3, \text{dim} = 1)$, which together with the clipping fulfills the premises two and three.

Finally, the mask is given by truncating the dummy system $\mathbf{g}_4 = [\mathbf{G}, \bar{\mathbf{c}}]$. Beware that only $\mathbf{g}_4$ is normalized along the system axis.

Application to protein dynamics. Since for proteins the final model is often expected to be invariant with respect to rotations and translations, internal coordinates are employed as input features. For Markov state modeling, the minimal heavy atom distance d_{ij} between residues i, j has been proven to be a good descriptor (56, 77). However, for interpretability, mask weights for each residue are preferable. Therefore, the mask is of size $\mathbf{G} \in \mathbb{R}^{R \times N}$ with the number of residues R. The input features are then scaled as $x_{ij} = G_i G_j \exp(-d_{ij})$.

Furthermore, a smoothing routine is implemented such that neighboring residues along the chain have similar importance weights. W windows of size B are placed along the chain with step size s. Each window has a trainable weight $\mathbf{g} \in \mathbb{R}^{W \times N}$. Consequently, the softmax function is taken along the window axis $\bar{\mathbf{g}} = \text{softmax}(\mathbf{g}, \text{dim=0})$. However, before applying the clipping as before the weight for each residue $\mathbf{g}_1 \in \mathbb{R}^{R \times N}$ is calculated as the product of all window weights the residue is part of (Fig. 6).

Data availability. The code that implements the presented models and reproduces the presented results can be found in our GitHub repository. The molecular dynamics data set of synaptotagmin C2A is available upon request. Restrictions apply to the availability of the villin data set, which were used under license for this study. Data are available from the authors upon request. The villin headpiece folding data is courtesy of D.E. SHAW research (76).

ACKNOWLEDGMENTS. We acknowledge financial support from Deutsche Forschungsgemeinschaft DFG (SFB/TRR 186, Project A12; SFB 1114, Projects C03 and A04; SFB 1078, Project C7, and RTG 2433), the European Commission (ERC CoG 772230 "ScaleCell"), the Berlin Mathematics center MATH+ (AA1-6 and AA1-10), the BMBF (Research center BIFOLD), the National Science Foundation (CHE-1738990, CHE-1900374, and PHY-2019745), the Welch Foundation (C-1570), and the Einstein Foundation Berlin. We further thank Manuel Dibak and Moritz Hoffmann (FU Berlin) for fruitful discussions.

1. JC Phillips, et al., Scalable molecular dynamics on cpu and gpu architectures with namd. *The J. chemical physics* **153**, 044130 (2020).
2. JW Vant, et al., Molecular dynamics flexible fitting: All you want to know about resolution exchange in *Protein Structure Prediction*. (Springer), pp. 301–315 (2020).
3. I Buch, MJ Harvey, T Giorgino, DP Anderson, G De Fabritiis, High-throughput all-atom molecular dynamics simulations using distributed computing. *J. chemical information modeling* **50**, 397–403 (2010).
4. P Eastman, et al., Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology* **13**, e1005659 (2017).
5. R Salomon-Ferrer, AW Gotz, D Poole, S Le Grand, RC Walker, Routine microsecond molecular dynamics simulations with amber on gpus. 2. explicit solvent particle mesh ewald. *J. chemical theory computation* **9**, 3878–3888 (2013).
6. MJ Abraham, et al., Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **1**, 19–25 (2015).
7. G Bussi, A Laio, P Tiwary, Metadynamics: A unified framework for accelerating rare events and sampling thermodynamics and kinetics. *Handb. Mater. Model. Methods: Theory Model.* pp. 565–595 (2020).
8. ST Tsai, Z Smith, P Tiwary, Sgoop-d: Estimating kinetic distances and reaction coordinate dimensionality for rare event systems from biased/unbiased simulations. *J. Chem. Theory Comput.* (2021).
9. C Liu, E Brini, A Perez, KA Dill, Computing ligands bound to proteins using meld-accelerated md. *J. Chem. Theory Comput.* **16**, 6377–6382 (2020).
10. R Nassar, et al., Determining protein structures using accelerated md simulations and noisy data. *Biophys. J.* **118**, 141a (2020).
11. A Perez, J MacCallum, KA Dill, Meld: Modeling peptide-protein interactions. *Biophys. J.* **104**, 399a (2013).
12. S Zhang, V Voelz, Improved estimates of folding stabilities and kinetics from multiensemble markov models. *Biophys. J.* **118**, 139a (2020).
13. JML Ribeiro, P Bravo, Y Wang, P Tiwary, Reweighted autoencoded variational bayes for enhanced sampling (rave). *The J. chemical physics* **149**, 072301 (2018).
14. C Schütte, A Fischer, W Huisinga, P Deuflhard, A Direct Approach to Conformational Dynamics based on Hybrid Monte Carlo. *J. Comput. Phys.* **151**, 146–168 (1999).
15. JH Prinz, et al., Markov models of molecular kinetics: Generation and validation. *J. Chem. Phys.* **134**, 174105 (2011).
16. WC Swope, JW Pitera, F Suits, Describing protein folding kinetics by molecular dynamics simulations: 1. Theory. *J. Phys. Chem. B* **108**, 6571–6581 (2004).
17. F Noé, I Horenko, C Schütte, JC Smith, Hierarchical analysis of conformational dynamics in biomolecules: transition networks of metastable states. *J Chem Phys* **126** (2007).
18. JD Chodera, et al., Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics. *J. Chem. Phys.* **126**, 155101 (2007).
19. NV Buchete, G Hummer, Coarse Master Equations for Peptide Folding Dynamics. *J. Phys. Chem. B* **112**, 6057–6069 (2008).
20. H Wan, VA Voelz, Adaptive markov state model estimation using short reseeding trajectories. *The J. chemical physics* **152**, 024103 (2020).
21. MK Scherer, et al., PyEMMA 2: A software package for estimation, validation and analysis of Markov models. *J. Chem. Theory Comput.* **11**, 5525–5542 (2015).
22. MP Harrigan, et al., Msmbuilder: Statistical models for biomolecular dynamics. *Biophys J.* **112**, 10–15 (2017).
23. RT McGibbon, et al., Mdtraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophys J.* **109**, 1528–1532 (2015).
24. W Humphrey, A Dalke, K Schulten, Vmd - visual molecular dynamics. *J. Molec. Graph.* **14**, 33–38 (1996).
25. G Perez-Hernandez, F Paul, T Giorgino, G D Fabritiis, F Noé, Identification of slow molecular order parameters for markov model construction. *J. Chem. Phys.* **139**, 015102 (2013).
26. A Ziehe, KR Müller, TDSEP — an efficient algorithm for blind separation using time structure in *ICANN 98*. (Springer Science and Business Media), pp. 675–680 (1998).
27. I Mezić, Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynam.* **41**, 309–325 (2005).
28. PJ Schmid, J Sesterhenn, Dynamic mode decomposition of numerical and experimental data in *61st Annual Meeting of the APS Division of Fluid Dynamics. American Physical Society*. (2008).
29. JH Tu, CW Rowley, DM Luchtenburg, SL Brunton, JN Kutz, On dynamic mode decomposition: Theory and applications. *J. Comput. Dyn.* **1**, 391–421 (2014).
30. F Noé, C Clementi, Collective variables for the study of long-time kinetics from molecular trajectories: theory and methods. *Curr. Opin. Struc. Biol.* **43**, 141–147 (2017).
31. S Klus, et al., Data-driven model reduction and transfer operator approximation. *J. Nonlinear Sci.* **28**, 985–1010 (2018).
32. GR Bowman, VS Pande, F Noé, eds., *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation.*, Advances in Experimental Medicine and Biology. (Springer Heidelberg) Vol. 797, (2014).
33. BE Husic, VS Pande, Ward clustering improves cross-validated markov state models of protein folding. *J. Chem. Theo. Comp.* **13**, 963–967 (2017).
34. FK Sheong, DA Silva, L Meng, Y Zhao, X Huang, Automatic State Partitioning for Multibody Systems (APM): An Efficient Algorithm for Constructing Markov State Models To Elucidate Conformational Dynamics of Multibody Systems. *J. Chem. Theory Comput.* **11**, 17–27 (2015).
35. M Weber, K Fackeldey, C Schütte, Set-free markov state model building. *J. Chem. Phys.* **146**, 124133, (2017).
36. GR Bowman, KA Beauchamp, G Boxer, VS Pande, Progress and challenges in the automated construction of Markov state models for full protein systems. *J. Chem. Phys.* **131**, 124101 (2009).
37. B Trendelkamp-Schroer, H Wu, F Paul, F Noé, Estimation and uncertainty of reversible markov models. *J. Chem. Phys.* **143**, 174101 (2015).
38. S Kube, M Weber, A coarse graining method for the identification of transition rates between molecular conformations. *J. Chem. Phys.* **126**, 024103 (2007).
39. Y Yao, et al., Hierarchical nyström methods for constructing markov state models for conformational dynamics. *J. Chem. Phys.* **138**, 174106 (2013).
40. K Fackeldey, M Weber, Genpcca – markov state models for non-equilibrium steady states. *WIAS Rep.* **29**, 70–80 (2017).
41. S Gerber, I Horenko, Toward a direct and scalable identification of reduced models for categorical processes. *Proc. Natl. Acad. Sci. USA* **114**, 4863–4868 (2017).
42. G Hummer, A Szabo, Optimal dimensionality reduction of multistate kinetic and markov-state models. *J. Phys. Chem. B* **119**, 9029–9037 (2015).

43. S Orioli, P Faccioli, Dimensional reduction of markov state models from renormalization group theory. *J. Chem. Phys.* **145**, 124120 (2016).
44. F Noé, H Wu, JH Prinz, N Plattner, Projected and hidden markov models for calculating kinetics and metastable states of complex molecules. *J. Chem. Phys.* **139**, 184114 (2013).
45. U Sengupta, M Carballo-Pacheco, B Strodel, Automated markov state models for molecular dynamics simulations of aggregation and self-assembly. *The J. chemical physics* **150**, 115101 (2019).
46. M Carballo-Pacheco, B Strodel, Advances in the simulation of protein aggregation at the atomistic scale. *The journal physical chemistry B* **120**, 2991–2999 (2016).
47. Q Qiao, GR Bowman, X Huang, Dynamics of an Intrinsically Disordered Protein Reveal Metastable Conformations That Potentially Seed Aggregation. *J. Am. Chem. Soc.* **135**, 16092–16101 (2013).
48. DA Silva, GR Bowman, A Sosa-Peinado, X Huang, A Role for Both Conformational Selection and Induced Fit in Ligand Binding by the LAO Protein. *PLoS Comput. Biol* **7**, e1002054 (2011).
49. U Sengupta, B Strodel, Markov models for the elucidation of allosteric regulation. *Philos. Transactions Royal Soc. B: Biol. Sci.* **373**, 20170178 (2018).
50. N Plattner, F Noé, Protein conformational plasticity and complex ligand-binding kinetics explored by atomistic simulations and Markov models. *Nat. Commun.* **6**, 7653 (2015).
51. CR Baiz, et al., A molecular interpretation of 2d ir protein folding experiments with markov state models. *Biophys. journal* **106**, 1359–1370 (2014).
52. S Olsson, H Wu, F Paul, C Clementi, F Noé, Combining experimental and simulation data of molecular processes via augmented Markov models. *Proc. Natl. Acad. Sci.* **114**, 8265–8270 (2017).
53. F Noé, F Nüske, A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Model. Simul.* **11**, 635–655 (2013).
54. RT McGibbon, VS Pande, Variational cross-validation of slow dynamical modes in molecular kinetics. *The J. Chem. Phys.* **142**, 124105 (2015).
55. H Wu, F Noé, Variational Approach for Learning Markov Processes from Time Series Data. *J. Nonlinear Sci.* (2019).
56. A Mardt, L Pasquali, H Wu, F Noé, Vampnets: Deep learning of molecular kinetics. *Nat. Commun.* **9**, 5 (2018).
57. W Chen, H Sidky, AL Ferguson, Nonlinear discovery of slow molecular modes using state-free reversible vampnets. *The J. chemical physics* **150**, 214114 (2019).
58. L Bonati, G Piccini, M Parrinello, Deep learning the slow modes for rare events sampling. *Proc Natl Acad Sci USA* **118**, e2113533118 (2021).
59. A Mardt, L Pasquali, F Noé, H Wu, Deep learning markov and koopman models with physical constraints in *Mathematical and Scientific Machine Learning*. (PMLR), pp. 451–475 (2020).
60. H Wu, A Mardt, L Pasquali, F Noe, Deep generative markov state models in *Advances in Neural Information Processing Systems*. pp. 3975–3984 (2018).
61. A Mardt, F Noé, Progress in deep markov state modeling: Coarse graining and experimental data restraints. *The J. Chem. Phys.* **155**, 214106 (2021).
62. KA Konovalov, IC Unarta, S Cao, EC Goonetilleke, X Huang, Markov state models to study the functional dynamics of proteins in the wake of machine learning. *JACS Au* (2021).
63. S Olsson, F Noé, Dynamic graphical models of molecular kinetics. *Proc. Natl. Acad. Sci.* **116**, 15001–15006 (2019).
64. T Hempel, et al., Independent Markov decomposition: Toward modeling kinetics of biomolecular complexes. *Proc Natl Acad Sci USA* **118**, e2105230118 (2021).
65. BO Koopman, Hamiltonian systems and transformations in hilbert space. *Proc. Natl. Acad. Sci. USA* **17**, 315–318 (1931).
66. KT Schütt, HE Sauceda, PJ Kindermans, A Tkatchenko, KR Müller, SchNet – A deep learning architecture for molecules and materials. *The J. Chem. Phys.* **148**, 241722 (2018).
67. K Schütt, O Unke, M Gastegger, Equivariant message passing for the prediction of tensorial properties and molecular spectra in *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, eds. M Meila, T Zhang. (PMLR), Vol. 139, pp. 9377–9388 (2021).
68. A Paszke, et al., Pytorch: An imperative style, high-performance deep learning library in *Advances in neural information processing systems*. pp. 8026–8037 (2019).
69. M Hoffmann, et al., Deeptime: A Python library for machine learning dynamical models from time series data. *ArXiv211015013 Math-Ph Physicsphysics Stat* (2021).
70. LR Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**, 257–286 (1989).
71. I Wolfram Research, Mathematica, Version 11.2.0 (2017).
72. AA Hagberg, DA Schult, PJ Swart, Exploring network structure, dynamics, and function using NetworkX in *Proceedings of the 7th Python in Science Conference*, eds. G Varoquaux, T Vaught, J Millman. (Pasadena, CA USA), pp. 11–15 (2008).
73. T Hempel, N Plattner, F Noé, Coupling of conformational switches in calcium sensor unraveled with local markov models and transfer entropy. *J. chemical theory computation* **16**, 2584–2593 (2020).
74. TC Südhof, Neurotransmitter release: The last millisecond in the life of a synaptic vesicle. *Neuron* **80**, 675–690 (2013).
75. JL Jiménez, et al., Functional Recycling of C2 Domains Throughout Evolution: A Comparative Study of Synaptotagmin, Protein Kinase C and Phospholipase C by Sequence, Structural and Modelling Approaches. *J. Mol. Biol.* **333**, 621–639 (2003).
76. K Lindorff-Larsen, S Piana, RO Dror, DE Shaw, How fast-folding proteins fold. *Science* **334**, 517–520 (2011).
77. MK Scherer, et al., Variational selection of features for molecular kinetics. *The J. Chem. Phys.* **150**, 194108 (2019).