

Deep Learning Protein Conformational Space with Convolutions and Latent Interpolations

Venkata K. Ramaswamy^{1,*}, Samuel C. Musson^{1,*}, Chris G. Willcocks^{2,‡} and Matteo T. Degiacomi^{1,†}

¹*Department of Physics, Durham University, Stockton Road, Durham DH1 3LE, United Kingdom*

²*Department of Computer Science, Durham University, Stockton Road, Durham DH1 3LE, United Kingdom*

 (Received 8 June 2020; revised 15 December 2020; accepted 26 January 2021; published 15 March 2021)

Determining the different conformational states of a protein and the transition paths between them is key to fully understanding the relationship between biomolecular structure and function. This can be accomplished by sampling protein conformational space with molecular simulation methodologies. Despite advances in computing hardware and sampling techniques, simulations always yield a discretized representation of this space, with transition states undersampled proportionally to their associated energy barrier. We present a convolutional neural network that learns a continuous conformational space representation from example structures, and loss functions that ensure intermediates between examples are physically plausible. We show that this network, trained with simulations of distinct protein states, can correctly predict a biologically relevant transition path, without any example on the path provided. We also show we can transfer features learned from one protein to others, which results in superior performances, and requires a surprisingly small number of training examples.

DOI: [10.1103/PhysRevX.11.011052](https://doi.org/10.1103/PhysRevX.11.011052)

Subject Areas: Biological Physics,
Computational Physics

I. INTRODUCTION

The shape of a protein determines its capacity of interacting, under given environmental conditions, with specific binding partners as different as ions, DNA, lipids, drugs, or other proteins. These tightly controlled interactions are key for life as we know it. Proteins should however not be considered as a single atomic arrangement, as their relationship with the environment (e.g., temperature, pressure, pH , binding to other molecules) results in specific conformational dynamics. These dynamics define a continuous conformational space, often conveniently subdivided into a discrete set of low-energy states and higher-energy transition paths between them. A full understanding of the function of any protein in an organism thus requires accurate knowledge of its conformational space. Proteins are often composed of thousands of atoms, that should in principle be associated with an enormous amount of possible arrangements. While only a small energetically favorable fraction of these arrangements is accessible [1], only a subset of these, those of lowest

energy, can be typically observed at atomic resolution by experimental techniques. In order to obtain a larger amount of conformational space samples, simulation techniques such as molecular dynamics (MD) can be exploited. These techniques iteratively generate new structures based on an initial, known atomic arrangement and a physical model of atomic interactions. However, biologically relevant conformational changes can sometimes take place in the timescale of milliseconds or higher (e.g., the full cycle of GroEL chaperonin from its closed to open state takes ~ 15 s [2]) which are beyond the reach of conventional MD.

Deep neural networks are able to learn continuous representations that capture the structure of a dataset. In particular, generative models such as variational autoencoders [3], generative adversarial networks [4], or Boltzmann generators [5] have been showing a remarkable ability to synthesize complex and sparse datasets. Generative neural networks create an internal model recapitulating example data, a model that can then be interrogated to generate new, plausible data samples. While most generative models produce new samples from an assumed prior distribution, recent architectures improve interpolations through additional adversarial components [6]. Many successful generative architectures utilize convolutional neural networks (CNNs) [7]. These are more computationally efficient than regular neural networks and have fewer parameters due to shared weights mimicking local connectivity in the visual cortex of the brain. Though a surfeit of applications of CNNs are in the fields of image, video, audio, and speech recognition [7], variants have also been applied to bioinformatics ranging from gene

*These authors contributed equally to this work.

†Corresponding author.

matteo.t.degiacomio@durham.ac.uk

‡Corresponding author.

christopher.g.willcocks@durham.ac.uk

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

expression regulation [8–10] and anomaly classification [11–13] to prediction of protein secondary structure [14,15] and protein folds [16–19].

We present a 1D CNN architecture [Fig. 1(a)], directly trainable with protein structures to build a model of their underlying conformational space. To improve the generalization capability of our network, we design a new loss function that leverages on, as a prior, knowledge of physical laws dictating atomic interactions. We enforce this physics-based loss function on the manifold between two protein conformations but outside of the known sampled conformational space. Our architecture and training approach lead to several significant advantages over conventional networks taking atomic coordinates [20] or molecular features as input. First, being fully convolutional, our architecture features a small number of parameters and is therefore easy to train. Second, it can handle input molecules with arbitrary numbers of atoms, enabling network training with different molecules, either simultaneously or via transfer learning. Third, it does not make assumptions on the distribution of data around observations used for training. We show that these features can enable the identification of biologically relevant intermediate protein configurations along plausible transition paths between known low-energy states.

II. RESULTS

MurD (UDP-N-acetylmuramoyl-L-alanine:D-glutamate ligase) [21] plays a key role in the peptidoglycan biosynthesis of almost all bacterial species by catalyzing the addition of D-glutamic acid to UDP-N-acetylmuramoyl-L-alanine. The protein consists of three globular domains, one of which undergoes a large-scale rearrangement (from open to closed state) triggered by substrate binding to activate the catalytic site. The open (PDB: 1E0D [22]) and closed (PDB: 3UAG [23]) states, as well as a few intermediates (PDB: 5A5E and 5A5F [24]), have been crystallized, providing key experimental evidence about the possible protein’s mode of action. MD simulations of the open and closed states, and of the transition between them, have been previously carried out [20], providing a useful dataset for our network training and its performance evaluation. Such extensive data and the importance of MurD as a potential antibacterial drug target [25,26] make this protein a particularly interesting test case for this study.

Here, we train our neural network with conformations of MurD open and closed states generated by MD (*training set*) and assess the network capacity of predicting a possible transition path between the two. We assess the quality of the predicted path in terms of its structural quality as well as matching with the closed-to-open MD simulation and

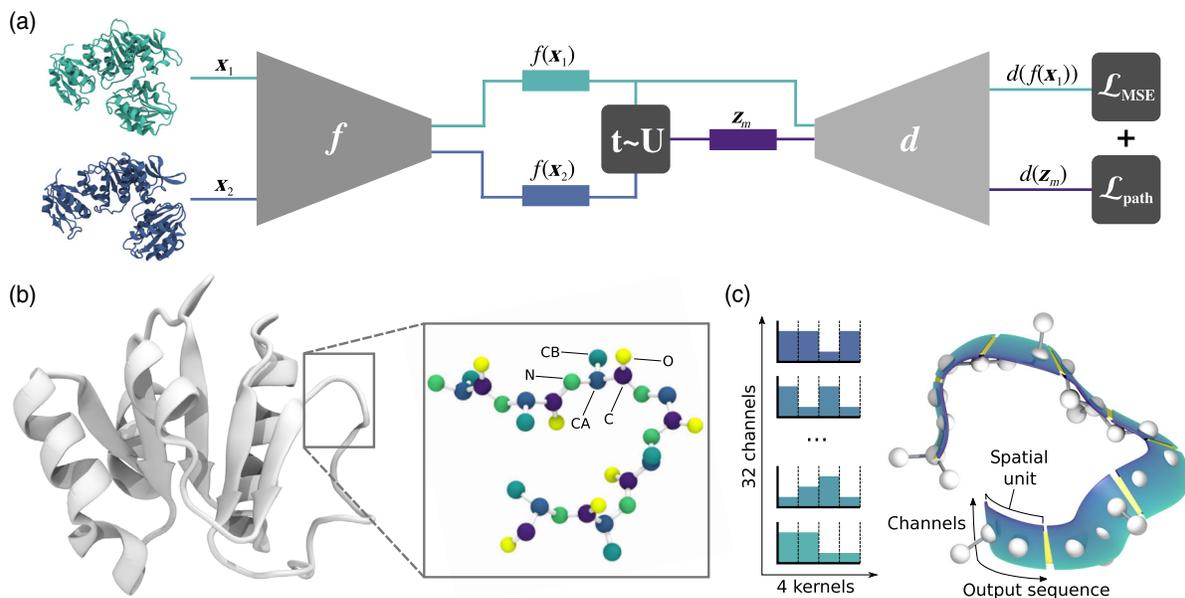


FIG. 1. Neural network design. (a) The generative architecture is composed of an encoder f and a decoder d , and is trained with a collection of protein conformations. The loss function couples a geometric term \mathcal{L}_{MSE} to ensure original and encoded-decoded structure are similar, and physics-based terms $\mathcal{L}_{\text{path}}$ to ensure that latent space interpolations z_m between any pair of conformations produce protein structures of low energy. (b) Protein atoms can be sorted in a list so that atoms that are adjacent in the list are also adjacent in the Cartesian space. The convolutional neural network operates on this list, that can be of arbitrary size. (c) The first 1D convolution layer learns $32 \times$ feature detectors, each with a kernel size of 4. The stride is set to 2; therefore the output sequence is half the input size for any input length. Each subsequent layer further reduces the spatial length of the molecule, warping the input such that it becomes progressively deeper and thicker (more ribbonlike) as well as more abstract.

available intermediate crystal structures, not provided for training. We then evaluate the capacity of our network trained on MurD to adapt to other proteins in a transfer learning scenario.

A. Force-field-based loss functions improve network accuracy

We first assessed whether training the neural network using information on the physical properties of proteins is beneficial. To this end, we trained it with five different loss functions featuring an increasing amount of physics-based constraints (Fig. 2 herein and Fig. S1 in Supplemental Material [27]). Each network was trained with conformations produced by the MD simulations of MurD closed and open states. To track the network training progress, we divided it into 200 parts termed epochs. At the end of every epoch, each network was challenged to predict a transition path by producing 20 intermediate structures corresponding to evenly spaced points along the straight line connecting the two states in the networks' latent space. To assess the quality of intermediate conformations throughout the training, we measured three different quantities: average distance of all bonds and angles from their expected equilibrium value (as per the AMBER ff14SB force field [28]), percentage of residues in favorable and outlier Ramachandran plot regions (assessment of dihedral angles

distributions), and discrete optimized protein energy (DOPE, an atomic distance-dependent statistical potential commonly used to assess protein structures) score [29].

MurD structures generated by the network trained without any physics-based constraints (using just the mean square error, DNN_{MSE}) could only poorly reproduce expected bond and angle equilibrium values. The average error of intermediate structures generated at training completion was equal to $\sim 22.9\% \pm 2.3$, with an average DOPE score of $-33017 (\pm 1005)$. Notably, the interpolation quality degraded the further the intermediate structure was from examples within the training set (errors of up to 26.6% and DOPE score reaching ~ -27642). Regions of the protein featuring the highest error were concentrated in loops and on the mobile domain of the protein [Fig. 2(b), DNN_{MSE}].

Coupling information on bonded (bonds, angles, dihedrals) and nonbonded (steric hindrance and electrostatics) potentials with MSE in the loss function improved the quality of the protein structures generated [Fig. 2(b), DNN_{Phys}]. In particular, the average DOPE score of intermediate structures generated at training completion was $-34810 (\pm 752)$, and the transition state conformations showed a maximum of -31601 , lower than what was measured on DNN_{MSE} . Overall, the residue-level DOPE score of DNN_{Phys} models was comparable with the profile of MurD crystal structure (PDB: 3UAG; see Fig. S2 in

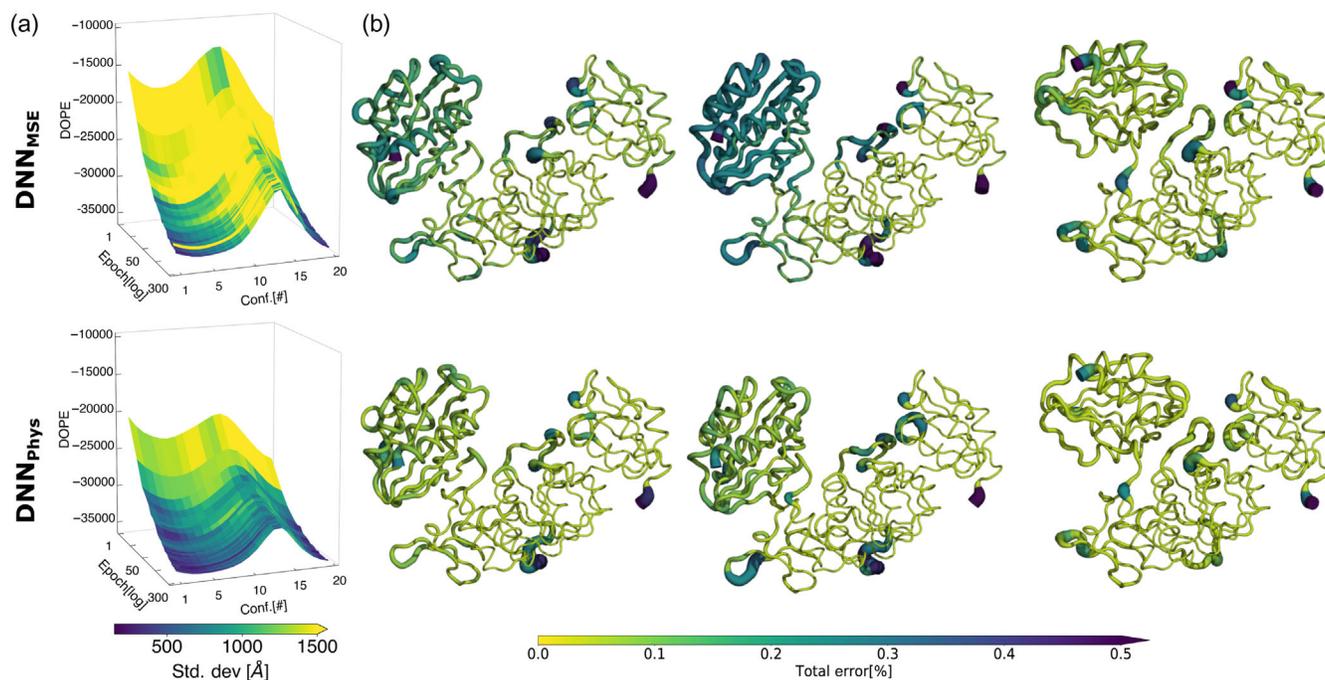


FIG. 2. Performance of the neural network trained with to minimize MSE alone or in conjunction with physical terms (labeled on the left). At every epoch, each network was asked to generate 20 conformations interpolating from MurD closed to open state. (a) At each epoch, we calculate the DOPE score of each conformation and report the mean values from the 10 repeats on the vertical axes with color corresponding to the standard deviation. Physics-based loss functions lead to interpolations with better structural quality reflected by lower DOPE scores. (b) The network-predicted protein conformations of open (left), intermediate (center), and closed (right) states at the last epoch, shown in sausage representation with the thickness and color corresponding to the percentage error at the residue level.

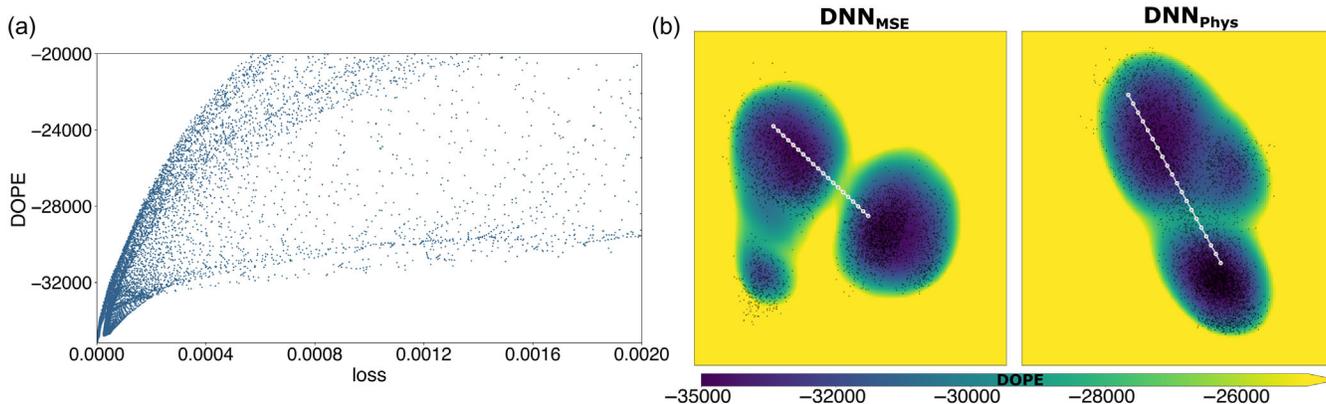


FIG. 3. Analysis of latent spaces of networks trained on conformations of MurD “open” and “closed” states. (a) The physics-based loss function used to train our neural network correlates with the DOPE score, making it a reasonable estimator of protein structural quality. (b) DOPE scores corresponding to the latent spaces of our neural network trained with two different loss functions. On the left, the network is trained to only minimize mean square error (MSE) between input and output structure; on the right, the loss function combines MSE and a physics-based loss function (DDN). Yellow regions indicate structures of poor quality, black points report on the projection in the latent space of all training examples, including the generated midpoints \mathbf{z}_m used to train DDN. The transition path predicted by the networks is projected as connected white dots onto their respective latent space. The latent space of the network based solely on MSE features two basins associated with closed and open states, separated by a region of poor quality models. In the network trained with both MSE and physics (DNN_{Phys}), the two states are connected by acceptable protein models. Also see Fig. S4 comparing the structures from the identified basins (Supplemental Material [27]).

Supplemental Material [27]). The network showed an increasingly better performance in the protein structural quality with increasing number of physics-based terms (Fig. S1 [27]). However, addition of the nonbonded component showed only a statistically insignificant improvement.

Evaluating Ramachandran distributions of amino acids in our models further highlighted how using physics-based restraints in DNN_{Phys} allow producing models of higher structural quality than DNN_{MSE} (Fig. S3 [27]). Models produced by DNN_{MSE} featured $87.6\% \pm 2.3$ of residues into Ramachandran favored regions and $4.3\% \pm 1.3$ outliers. Models produced by DNN_{Phys} featured a higher percentage of residues in favored regions ($93.4\% \pm 1.3$) and only $1.8\% \pm 0.7$ outliers, all located close to the acceptable ranges.

To obtain a comprehensive view of the protein conformational space encoded in our neural networks, we calculated the DOPE score of protein models generated by regularly sampling the networks’ latent space (Fig. 3). The latent space of the network trained solely with MSE featured two distinct regions, associated with MurD open and closed state, separated by a region of unacceptable quality (high DOPE score). In the network trained with our physics-based loss function, the whole conformational space appears near convex, and all structures located between the “closed” and “open” basins had a low DOPE score (Fig. S5 of Supplemental Material [27]). This was possible because the latter network is also trained to attempt minimizing the energy of random midpoints between training examples. Overall, training our 1D CNN with a

loss function featuring a combination of MSE, bonded and nonbonded terms yielded interpolations of good structural quality through conformations not represented within the training set.

B. Neural network predicts a possible state transition path

The switch of MurD between closed and open conformations involves the rigid-body rearrangement of one domain (residues 299–437) with respect to the rest of the protein (residues 1–298). We can readily characterize this movement by tracking the position of the center of mass of this domain with respect to its connection to the rest of the protein and reporting it in spherical coordinates [Fig. 4(a)]. Describing the closed and open MurD MD simulations according to this metric reveals that the conformations of these two states are clearly distinct [two light green regions in Fig. 4(b)]. The MD simulation of MurD switching from closed to open state (hereon *test set*) follows an irregular path: first, a concerted increase in elevation and azimuth opens the domain, leading to conformations closely resembling the crystal structure of the intermediates [root mean square deviation (RMSD) of secondary structure elements equal to 1.16 and 1.12 Å versus PDBs 5A5E and 5A5F, respectively], then an increase in azimuth and radius leads the domain to its final equilibrium position. This two-step movement in the azimuth-elevation space is associated with a straight line within the latent space described by our network. As the neural network is trained to minimize the energy of any midpoint between any two states, we consider conformations generated by a straight line in

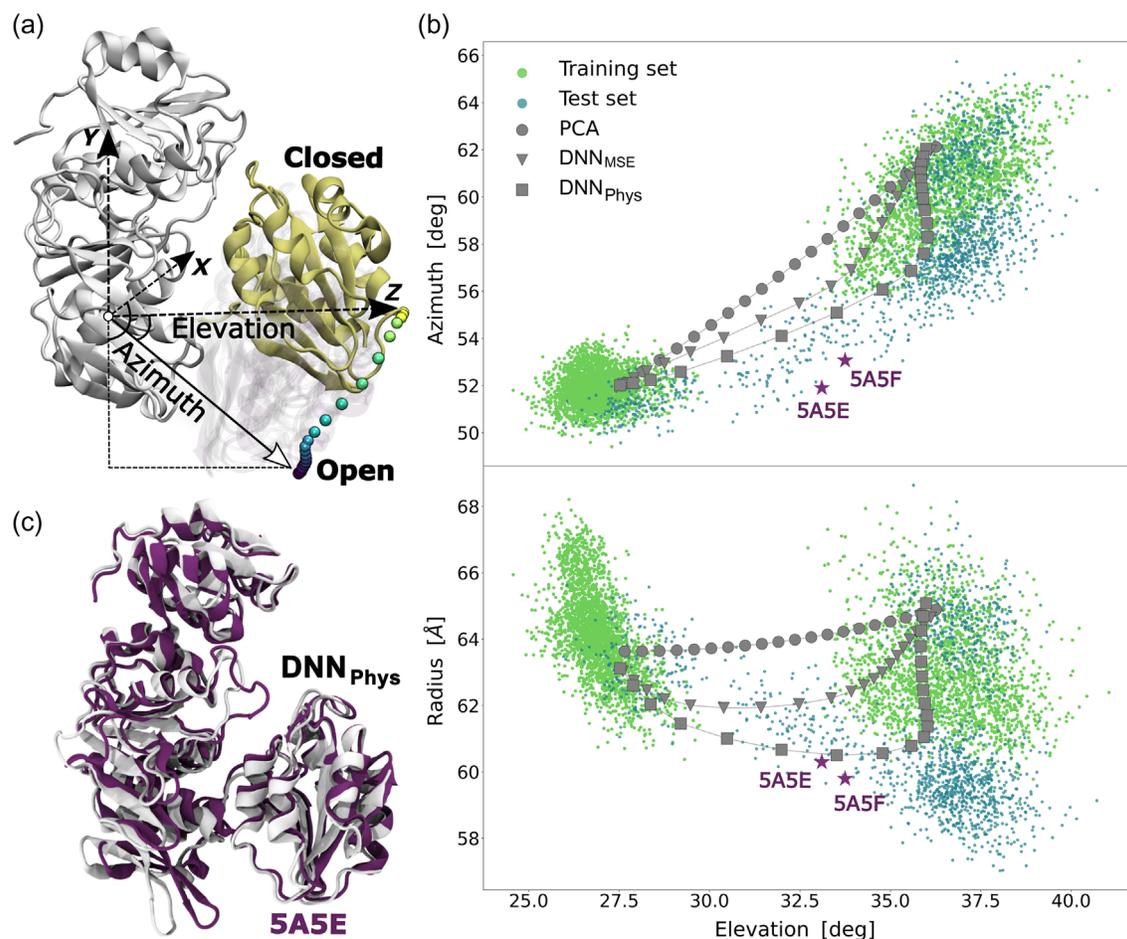


FIG. 4. State transition path prediction by the neural network. (a) Side view of MurD showing the transition of its mobile domain from the closed to open state (with the time evolution marked as beads colored from yellow to violet). (b) The conformational change of MurD can be described in terms of spherical coordinates between its three domains (see Methods). We report the opening angle of each conformation in the training set (light green), test set (dark green), intermediate crystal structures (palatinate stars), as well as interpolations generated by PCA (gray circles), purely geometry-based neural network (DNN_{MSE} , gray triangles), and neural network combining geometry and physics (DNN_{Phys} , gray squares). The interpolation produced by DNN_{Phys} best reproduces the path described by the test set and transits in the vicinity of intermediate crystal structures. (c) Superimposition of intermediate MurD conformation 5A5E (in palatinate) and an intermediate conformation generated by the neural network (in white), with a RMSD of 1.3 Å.

the latent space as an approximation of a possible transition path. To illustrate that methods relying on purely geometric interpolations would be unable to predict such transition path, we generated an interpolation exploiting a principal components analysis (PCA) of MurD conformations. Conformations generated as regularly spaced linear combinations of the two main eigenvectors trace a uniform circular trajectory, far from what was observed in the test set. Considering more than two eigenvectors also has no significant improvement in the interpolations generated by PCA (Fig. S6 [27]).

The transition path predicted by the neural network trained solely based on MSE is less uniform, but yet only poorly recapitulates the test set. The interpolation produced by the network trained considering both MSE and physics-based terms traces instead a path closely resembling the test

set. The additional physics-based terms in the loss function not only helped in generating structures with correct bond lengths, angles, and dihedrals, but also prevented an interpolation that would have required the protein domains to slightly compenetrare. Remarkably, one of the interpolated conformations had a backbone RMSD of 1.28 Å from the intermediate crystal structure (PDB: 5A5E), a quantity comparable to that of the test set (Fig. S7 [27]).

Thus, our results show that in this application our 1D CNN trained with a combination of MSE and physics-based terms was capable of correctly identifying a possible transition path between two distinct states.

C. Transfer learning improves convergence

Training a neural network on a small dataset can be facilitated if the network is pretrained on a similar, larger

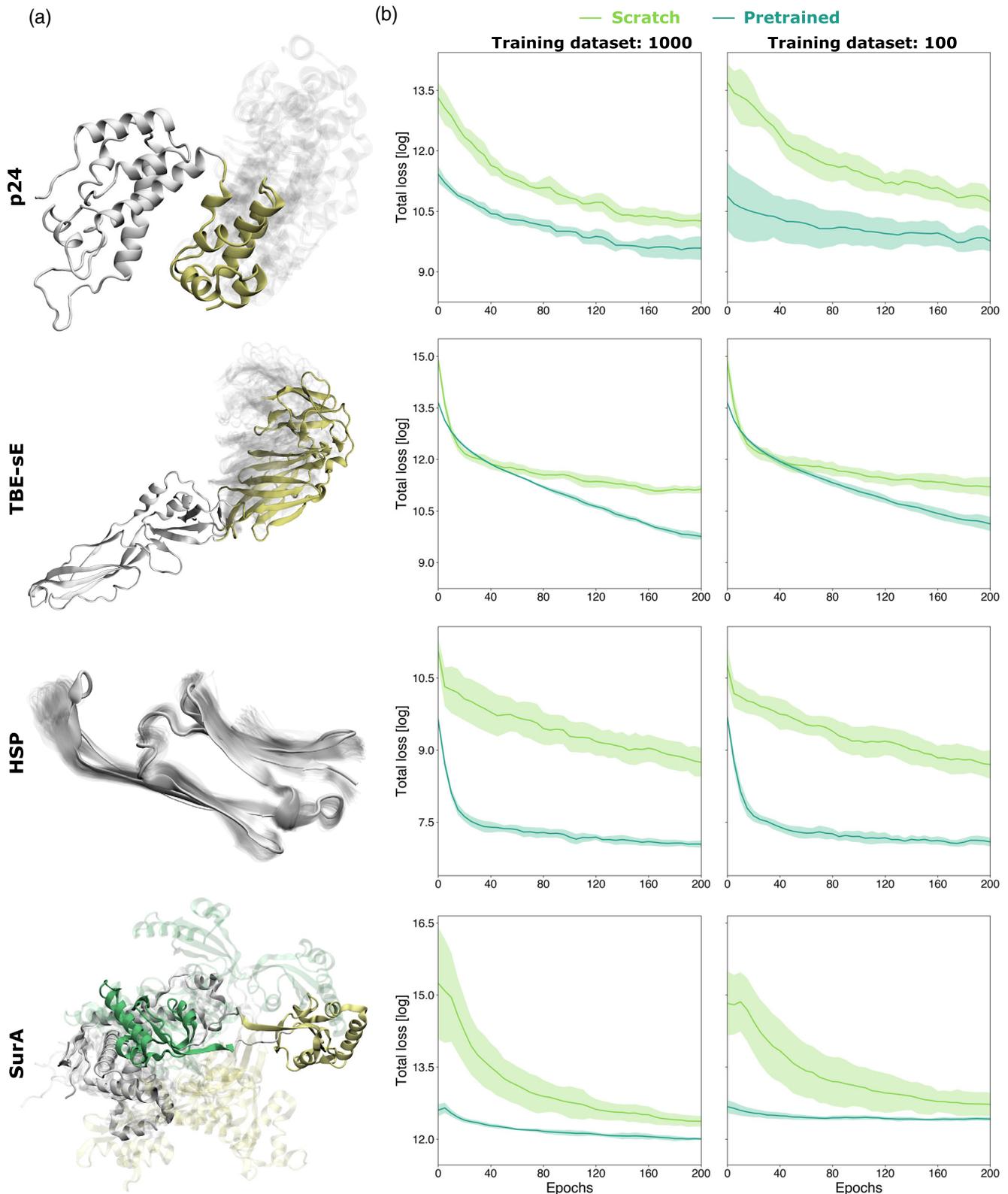


FIG. 5. Transfer learning of our best network trained on MurD to four different proteins. (a) Side view of p24, TBE-sE, HSP, and SurA showing example alternative conformations with transparency and the highly mobile domains of p24, TBE-sE, and SurA in color (yellow and green). (b) Moving average (window of 3) mean (solid line) and standard deviation (shaded region) of the total loss (log scale) comparing the performance of the pretrained network with its counterpart trained from scratch using a training dataset of size 1000 and 100, all using the same loss function featuring both MSE and physics-based terms.

dataset, a process called transfer learning. The convolutional nature of our network enables this training approach, as its architecture (and thus the number of its parameters) is independent of the number of atoms in the dataset under study. To assess the transferability of our trained network, we leveraged the MD simulations of four different proteins: the HIV-1 capsid protein (monomeric p24, PDB: 1E6J [30]), the tick-borne encephalitis virus envelope protein E (TBE-sE, PDB: 1SVB [31]), the small heat shock protein α B crystallin (HSP, PDB: 2WJ7 [32]), and the periplasmic chaperone SurA (SurA, PDB: 1M5Y [33]). These proteins differ from MurD in mass (24, 43, 10, and 45 kDa versus 47 kDa, respectively) as well as fold and dynamics [Fig. 5(a)]. p24 consists of two rigid domains connected by a flexible linker, TBE-sE is highly elongated and features several long loop regions (e.g., residues 73–89 and 146–160), HSP is small and features only local fluctuations, whereas SurA is a multidomain protein featuring broad and highly complex dynamics.

We randomly sampled each simulation dataset to produce multiple training sets featuring 1000 and 100 representative structures (see Sec. IV). For each resulting training set, we then trained our neural network in two different ways: from scratch and by initializing its parameters with those of the best network trained on the MurD dataset using our physics-based loss function. Each neural network was assessed according to its capacity of interpolating between the two most different conformations (i.e., largest RMSD) in its training set.

In all the cases, the networks having transferred parameters featured a mean loss starting from smaller values, and dropping faster. Networks trained with 1000 examples featured only a marginally lower mean loss than those trained with 100 examples [Fig. 5(b)]. Furthermore, transfer learning yielded latent spaces with overall lower DOPE scores. In all cases but SurA, negative DOPE scores were observed in regions extending slightly beyond the front defined by the training set (Fig. S8 [27]). SurA was the only protein featuring DOPE landscapes without low and well-defined minima. As their associated loss functions were larger than in all other cases and had not converged yet, we trained four neural networks (initialized from scratch and via transfer learning, trained with 100 and 1000 examples) 10 times longer. Their loss function and DOPE profiles kept dropping, indicating that learning complex dynamics requires longer training (Fig. S9 [27]).

To characterize the ability of our neural networks to generate structures unseen during training, we examined those exposed to examples of the protein p24. As demonstrated by an available crystal structure (PDB: 3MGE), this protein assembles into a circular homohexamer via a large-scale rearrangement of its two globular domains. The structure in the p24 unbound simulation most closely resembling the bound state has a RMSD of 3.7 Å from it. We asked each trained network to encode and decode the

bound state structure (unseen during training) and compared the RMSD of the resulting generated model to the input.

Neural networks trained with 100 examples performed equivalently (RMSD equal to 8.23 ± 1.72 Å for those trained from scratch, and 8.32 ± 1.65 Å for those pre-trained). This poor performance indicates that training with only 100 examples yields neural networks that, although already capable of generating low-energy protein conformations, have limited generalization capabilities. Using 1000 training examples led to substantially better results, with pretrained networks outperforming those trained from scratch (4.69 ± 0.15 Å the former, 5.22 ± 0.57 Å the latter). Since these RMSDs are still higher than that of the best available training example, we systematically sampled the networks' latent spaces. We found regions where generated structures featured RMSDs from the bound state marginally lower than any available example (~ 3.6 Å when training with 1000 examples). Remarkably, these regions were located beyond the front of training structures, meaning that most suitable bound state conformations could be found via local extrapolation (Fig. S10 [27]).

Overall, these results demonstrate that our network can be trained with proteins of arbitrary size, and indicate that transfer learning enables faster training convergence, even when training data are limited.

III. DISCUSSION AND CONCLUSION

Conformational dynamics are an intrinsic property of any protein, their magnitude depending on the required biological function. Several computational methods have been designed to sample the protein conformational landscape and predict existing states. Methods leveraging MD coupled with enhanced sampling methods, while helpful, are still limited by the timescales at which the transitions occur and the associated high demand for computational power. We have designed and trained a generative neural network with collections of protein conformations as a means to predict plausible intermediates between any of them. While here the network has been trained with structures from MD simulations, any source of structural information can in principle be used. Since as little as 100 structures were sufficient to produce physically correct models with our trained network, this opens the door to directly leveraging collections of experimentally determined atomic structures.

Generative networks [3,4,6] are effective when interpolating between existing data but, unless additional information is provided, their extrapolation capabilities are typically insufficient to generate plausible molecular structures. The additional difficulty when training a neural network with protein atomic coordinates is associated with the very high number of degrees of freedom to be handled

(often > 1000 individual x , y , and z coordinates), making the training process arduous. To overcome these limitations, we have designed a 1D CNN architecture and introduced physics-based terms in the loss function. The convolutional architecture, already commonplace for image and text processing as well as bioinformatics tasks, is associated with a smaller number of trainable parameters and is independent of the number of degrees of freedom (and thus atoms) within the training set. Our physics-based terms in the loss function directly constrain learning of protein structures obeying essential force-field potentials. To further facilitate the training process, we have introduced a warping on the nonbonded potential terms (see Methods), leading to better gradients encouraging convergence in otherwise high Lipschitz constant conditions. To test our network we have purposely designed a set of hard tasks, by selecting the most different conformations from protein simulations featuring different ranges of conformational changes, and asking the neural network to generate suitable interpolations by leveraging only a two-dimensional latent space, as well as a limited number of training examples.

Challenged with the flexible protein MurD, our network revealed a surprising transition path between two distinct states, closed and open. Remarkably, conformations along this path were both physically plausible and compatible with an existing MD simulation featuring a closed-to-open transition as well as with two crystal structures of MurD locked in intermediate conformations. While the MSE term in the loss function was key for the network to learn the global protein shape, the addition of physics-based terms was crucial to generating low-energy conformations, associated with a low DOPE score. The bonded terms (bonds, angles, dihedrals) helped fine-tuning local atomic arrangements, whereas the nonbonded ones (electrostatics [34] and Lennard-Jones [35]) were a significant player in identifying a suitable transition path within the conformational space. The structures produced by our new network resulted to be both of lower energy and greater biological relevance than those obtained by PCA-based interpolations or by a neural network trained solely according to a MSE-based loss function [Figs. 4(b) and 4(c)]. By analyzing the loss function values within the two-dimensional latent space of our physics-based neural network, we observed a clearly defined and near-convex minimum, despite the network having been trained with distinct conformational states [Fig. 3(b)].

An attractive feature of our convolutional network is that it can be trained with conformations of proteins of arbitrary amino acid sequence. This also enables transfer learning, whereby a pretrained network is repurposed to tackle a new though related task, expected to lead to improved generalization and faster network optimization. In this context, we noted that different proteins still share common features (e.g., typical bonds length and angles, as well as the same sequence of atoms in the backbone), that should not be

relearned. We transferred the network trained on MurD to four different proteins (HIV-1 capsomer protein p24, tick-borne encephalitis virus envelope protein E, α B crystallin small heat shock protein, and periplasmic chaperone SurA), for which we provided only a limited number of training examples (as low as 100). Remarkably, after few epochs the total error associated with structures generated by the pretrained network dropped lower than that of structures produced by the network trained from scratch.

A detailed analysis of generated protein structures indicated that our neural network may produce suboptimal loop regions when these are highly flexible and the protein movement is dominated by larger domain-level conformational changes. Thus, we expect our neural network to perform best with folded proteins, with atoms featuring correlated movements. The intermediate structures generated by our neural network will feature overall low energy according to a subset of typical terms associated with molecular structures, namely their bond, angles, dihedrals, van der Waals interactions and electrostatic interactions. Still, the energy of predicted intermediates will not be an accurate estimation of energy barriers a conformational change is associated with. Thus, while capable of local extrapolation, the network should not be expected to predict new, completely unseen states. Nevertheless, knowledge of possible transition paths can provide guidance for the definition of appropriate reaction coordinates or collective variables in MD-based enhanced sampling schemes. Furthermore, as the neural network effectively transforms a discrete collection of protein conformations into a “conformational continuum,” it can find applications in flexible protein-protein docking scenarios where, under the conformational selection paradigm, the ability of fine-tuning protein conformations to maximize their compatibility with a binding partner is desirable [20].

In summary, we have designed a new architecture with physics-based loss functions that significantly improve the synthesis of protein atomic structures in different conformational states. We have shown that biologically relevant transition paths can be predicted when synthesized interpolations must respect physical laws. The estimated transition path retains a surprising resemblance to the ground truth, and even more surprisingly this is achievable with only a small number of training examples. Further, our findings show that, through our fully convolutional architecture, we can transfer features learned from one protein to another, indicating the possibility of simultaneously training with conformations of multiple proteins, toward a network trained with the whole proteome.

IV. METHODS

A. Network architecture

Proteins are defined by their amino acid sequence, and each sequence maps onto an ensemble of possible three-

dimensional atomic arrangements (conformations). The space of possible conformations associated with a specific sequence may be extremely reduced for a protein taking a single well-defined state, or broad for a flexible protein capable of interconverting between multiple states. As the proteome is vast and many proteins are resistant to most forms of experimental interrogation, only a relatively small subset of proteins have had at least one of their possible conformations revealed at atomic resolution. Furthermore, as the techniques characterizing molecular structures typically report on low-energy conformations, transition states are undersampled proportionally to their associated energy barrier.

From a machine-learning perspective, we can define the entire proteome as a distribution $p_d(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^{3 \times n}$ is a protein, and $m_d(\mathbf{x})$ a collection of conformations of a specific protein experimentally (e.g., nuclear magnetic resonance spectroscopy, x-ray crystallography) or computationally (e.g., Monte Carlo or MD simulations) determined. Here, vector \mathbf{x} comprises of all the coordinates (\mathbb{R}) of a protein made of n atoms. We wish to learn a low m -dimensional embedding $f: \mathbb{R}^{3 \times n} \rightarrow \mathbb{R}^m$ that maps proteins onto the latent space, where sampling any point $\mathbf{z} \in \mathbb{R}^m$ and taking the inverse f^{-1} yields a continuous space of physically plausible molecular structures. However, as the expected observations $\mathbf{x} \sim p_d(\mathbf{x})$ and $\mathbf{x} \sim m_d(\mathbf{x})$ relax into a subset of conformations [36], the behavior at the valley regions on the manifold (maxima in the energy landscape) is difficult to capture explicitly from the observations.

Let $f(\mathbf{z}|\mathbf{x}; \theta)$ be an encoder function with parameters θ , and $d(\hat{\mathbf{x}}|\mathbf{z}; \theta)$ be a decoder function that approximates the inverse f^{-1} accordingly. The conventional approach is a simple reconstructive autoencoder that minimizes the mean squared error loss \mathcal{L}_{MSE} , where

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{\mathbf{x} \sim p_d(\mathbf{x})} [\|d(f(\mathbf{x})) - \mathbf{x}\|^2]. \quad (1)$$

This follows the geometry and probability distribution from which the dataset was collected and therefore fails to generalize, especially at undersampled regions associated with transition states.

Proteins undergo conformational changes following lower-energy paths in their energy landscape, where transition states are expected to be saddle points. The protein's expected energy, as determined by its atomic interactions, can be expressed as a loss function such as $\Psi(x, y) = \|x - y\|^2$, defined as

$$\mathcal{L}_{\text{Phys}} = \mathbb{E}_{\mathbf{x} \sim p_d(\mathbf{x})} [\Psi(d(f(\mathbf{x})))], \quad (2)$$

where Ψ evaluates the energy of the decoded structure. However, as with the naive autoencoder, this also fails to generalize at regions far from conformations provided as example. In principle, it is possible to enforce physics to be respected in regions outside the known conformational

space, for example, with a Gaussian prior in the latent encoding. However, this makes an assumption on the distribution of the latent space.

Any midpoint along the geodesic (i.e., shortest path on the learned manifold) between any two protein conformations $(\mathbf{x}_1, \mathbf{x}_2) \sim m_d(\mathbf{x})$ will also be a protein of same connectivity and composition. Assuming a degree of convexity of the latent space, we can enforce our physics-based loss function at random midpoints between \mathbf{x}_1 and \mathbf{x}_2 , sampled linearly from a uniform distribution:

$$\mathcal{L}_{\text{path}} = \mathbb{E}_{(\mathbf{x}_1, \mathbf{x}_2) \sim m_d(\mathbf{x}), t \sim U} [\Psi(d(\mathbf{z}_m))], \quad (3)$$

where $\mathbf{z}_m = (1 - t)f(\mathbf{x}_1) + tf(\mathbf{x}_2)$ are midpoints in the latent space between two protein conformations. This enables physical characteristics to be enforced on the manifold between two points, but outside of the known sampled conformational space.

We note that, in principle, this approach allows for the training of a neural network with m_d simulations of multiple proteins, whereby pairs (to interpolate between) are picked from the same simulation.

Putting this together, we can assemble the final loss as a weighted sum of the geometric term and the path term (which itself is comprised of various individually weighted physics terms):

$$\mathcal{L} = \alpha \mathcal{L}_{\text{MSE}} + \beta \mathcal{L}_{\text{path}}. \quad (4)$$

B. Masked bonded loss

In molecular mechanics a functional abstraction, termed a force field, is used to describe the potential energy $V(\mathbf{x})$ for a molecular system as a function of its atomic coordinates. The exact functional form of a force field can vary, but in their basic embodiment they can be described as a sum of individual energy terms:

$$\Psi(\mathbf{x}) = V_{\text{bonds}} + V_{\text{angles}} + V_{\text{dihedrals}} + V_{\text{nonbonded}}. \quad (5)$$

We define our physics-based loss to be functionally identical to the AMBER force field, thus compelling the potential energy of intermediate structures to be minimized. This would enforce physics to be respected at all points along the geodesic between two protein conformations and constrain the intermediates to follow a locally minimum energy path. We have developed a PyTorch implementation of the AMBER force field allowing us to both utilize graphical processing unit (GPU) acceleration and enable backpropagation through the physics loss.

In the AMBER force field, bonds are represented as a harmonic potential about the equilibrium bond distance r_0 :

$$V_{\text{bonds}} = \sum_{\text{bonds}} k_b (r - r_0)^2, \quad (6)$$

where k_b is force constant dependent on the specific type of bonded atoms. Force constants and equilibrium distances are taken directly from the AMBER ff14SB parameter sets.

Atoms in a protein structure file are listed residue by residue, so the positions of atoms throughout the list are spatially coherent, whereby atoms involved in a common bond or angle are most likely adjacent in the list. We can efficiently utilize GPUs by calculating \mathbf{r}^d such that $r_i^d = \|\mathbf{u}_i^d\|_2 = \|\mathbf{x}_i - \mathbf{x}_{i+d}\|_2$, then we can calculate the bond potential with

$$V_{\text{bonds}} = \sum_d \mathbf{k}_b (\mathbf{r}^d - \mathbf{r}_0)^2 \quad (7)$$

for all relevant d . We can mask any values in \mathbf{r}^d that do not correspond to bonded atoms by setting the corresponding element in \mathbf{k}_b^d to zero.

The AMBER force field represents angles similarly:

$$V_{\text{angles}} = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2, \quad (8)$$

where θ is the angle between two bond vectors \mathbf{u}^{d_1} and \mathbf{u}^{d_2} . We calculate the angle potential as

$$V_{\text{angles}} = \sum_{d_1, d_2} \mathbf{k}_\theta (\theta^{d_1, d_2} - \theta_0)^2, \quad (9)$$

where

$$\theta_i^{d_1, d_2} = \arccos \left(\frac{\mathbf{u}_i^{d_1} \cdot s^{d_2} \mathbf{u}_{i+d_1}^{d_2}}{\|\mathbf{u}_i^{d_1}\| \|\mathbf{u}_i^{d_2}\|} \right), \quad (10)$$

where for the angle between atom indices i , j , and k we can determine d_1 and d_2 as $j - i$ and $j - k$. The factor s^{d_2} is 1 if $j < k$ or -1 if $j > k$.

Dihedral angles are represented in the AMBER force field as

$$V_{\text{dihedrals}} = \sum_{\text{dihedrals}} V_n [1 + \cos(n\phi - \gamma)], \quad (11)$$

where V_n , n , ϕ , and γ represent the barrier, periodicity, torsion angle, and phase, respectively. We implement this as

$$V_{\text{dihedrals}} = \sum_n \sum_{d_1, d_2, d_3} V_n [1 + \cos(n\phi^{d_1, d_2, d_3} - \gamma)], \quad (12)$$

where

$$\phi_i^{d_1, d_2, d_3} = \text{atan2}(\mathbf{u}_2 \cdot ((\mathbf{u}_1 \times \mathbf{u}_2) \times (\mathbf{u}_2 \times \mathbf{u}_3)), \|\mathbf{u}_2\| ((\mathbf{u}_1 \times \mathbf{u}_2) \cdot (\mathbf{u}_2 \times \mathbf{u}_3))) \quad (13)$$

$$\mathbf{u}_x = s^{d_x} \mathbf{u}_i^{d_x} \quad (14)$$

The use of the atan2 function eliminates the possibility of a division by zero, resulting in infinite gradients.

C. Warped nonbonded loss

The nonbonded potential $V_{\text{nonbonded}}$ consists of a sum of two terms, describing van der Waals and electrostatic interactions. $V_{\text{nonbonded}}$ is evaluated for any pair of atoms not involved in a mutual bond, angle, or dihedral. Given \mathbf{r} , the distance between two atoms, the van der Waals interactions are described as in the AMBER force field by a 12-6 Lennard-Jones potential V_{LJ} :

$$V_{\text{LJ}} = \sum_i^{N-1} \sum_{j=i+1}^N \left[\left(\frac{A_{ij}}{r_{ij}^{12}} \right) - \left(\frac{B_{ij}}{r_{ij}^6} \right) \right], \quad (15)$$

$$A_{ij} = \epsilon_{ij} R_{ij}^{12} \quad \text{and} \quad B_{ij} = 2\epsilon_{ij} R_{ij}^6, \quad (16)$$

$$R_{ij} = 0.5(R_{\min, i} + R_{\min, j}), \quad (17)$$

$$\epsilon_{ij} = \sqrt{\epsilon_{ii} \cdot \epsilon_{jj}}, \quad (18)$$

where $R_{\min, x}$ and ϵ_{xx} are the van der Waals radius and well depth of the potential for an atom type x , as defined within the AMBER parameter set. Equations (17) and (18) show the Lorentz-Berthelot mixing rules used to obtain cross terms between different i and j . Since side-chains atoms (except $C\beta$) are not used in the network training, only the repulsive term of the Lennard-Jones is applied. This is to avoid the protein structure to pack excessively, filling the voids left by missing side-chain atoms.

Thus, here we only retain the repulsive component of the Lennard-Jones potential:

$$V_{\text{LJ}} = \sum_i^{N-1} \sum_{j=i+1}^N \left[\left(\frac{A_{ij}}{r_{ij}^{12}} \right) \right]. \quad (19)$$

Electrostatic interactions are described by a Coulombic potential V_C :

$$V_C = \sum_i^{N-1} \sum_{j=i+1}^N \left[k_e \frac{q_i q_j}{\epsilon_0 r_{ij}} \right], \quad (20)$$

where q_x is the charge on atom x , ϵ is the dielectric constant, and k is the Coulomb force constant. As $\lim_{r \rightarrow 0} V_{\text{nonbonded}}(r) = \infty$, the gradient descent becomes unstable at short interatomic distances. Clamping \mathbf{r} causes the gradients to get stuck in the corners of the hypercube. Therefore, we approximate the nonbonded potentials by warping the input space $\mathbf{r}' \approx \mathbf{r}$ piecewise with an exponential. This is achieved equivalently and efficiently with the exponential linear unit (ELU) [37] intrinsic activation function for some offset constant k , where

$$\mathbf{r}' = \text{ELU}(\mathbf{r} - k, \alpha = 1) + k, \quad (21)$$

such that the potentials now tend to high positive or negative values, without significantly altering the profile of the potential well (Fig. S11 in Supplemental Material [27]). We choose $k = 1.9$ for V_{LJ} and $k = 0.4$ for V_C , giving large positive or negative y intercepts for the expected upper and lower bounds for A_{ij} and $q_i q_j$ accordingly.

D. Implementation

Atoms (points) of a molecular structure [such as in PDB file, the standard representation for molecular 3D structure data, Fig. 1(b)] can be sorted in a list so that covalently connected atoms appear close to each other.

The 1D CNN architecture was implemented in PyTorch with 12 layers (6 in the encoder and 6 in the decoder components). The first layer has 3 input channels (for each atom’s 3D coordinates) and 32 output channels, where subsequent layers of depth t have $\lceil 32 \times 1.5^t \rceil$ input channels with batch normalization and rectified linear unit activations. Convolutions have size 4 kernels with strides of 2 and padding of 1, halving the spatial dimension each layer. This means the molecule becomes like a progressively thicker but shorter “ribbon” whose activations correspond to more deep and abstract characteristics of the molecule, Fig. 1(c). The latent space was fixed to two dimensions using a 1D adaptive mean pooling layer to handle arbitrary length input sequences.

We found that increasing the dimension of the latent space, adding an adversarial discriminating component, or adding residual layers led to only negligible generalization improvements. The total number of parameters is 11 892 767 optimized using Adam [38] with a learning rate of $1e^{-4}$ and weight decay of $1e^{-5}$. We used a batch size of 5 for 200 epochs and 1000 optimization steps per epoch for the training of MurD. Training the model takes ~ 7 h on an NVIDIA TITAN Xp graphics card. A larger batch size allows a higher utilization of the GPU, so for transfer learning MurD was retrained with a batch of 16 for 200 epochs and 500 optimization steps per epoch and retrained on the other structure sets at 100 optimization steps per epoch. The decreased number of optimization steps has no impact on training other than to increase the rate at which the network performance was tested and decrease the total allowed training time.

For the results given for MurD we repeated training of the neural network 10 independent times with random initial weights; all values reported (loss functions mean and standard deviation, as well as the DOPE score of resulting interpolated structures) are the average of these 10 repeats. For transfer learning to the other structure sets, we repeated training of the neural network 10 independent times, 5 with random initial weights and 5 retaining the weights from networks pretrained on MurD. The weights α and β in Eq. (4) were controlled at each step to keep the magnitude

of \mathcal{L}_{MSE} 10 times greater than $\mathcal{L}_{\text{path}}$, thus putting an order of magnitude greater emphasis on accurately reconstructing known structures as generating low-energy intermediate structures.

E. Datasets

MurD has been crystallized in its open (PDB: 1E0D [22]) and closed (PDB: 3UAG [23]) states, as well as in intermediates between the two (PDB: 5A5E and 5A5F [24] with a backbone RMSD of secondary structure elements equal to 1.12 Å between them). The difference between these states comes from large-scale conformational change of one of its three globular domains (residues 299–437) caused by substrate binding. Conformations from MD simulations of the closed and open states performed by some of the authors [20] were used here as the training dataset (4420 conformations in total comprising of 2507 and 1913 from closed and open simulations, respectively). In order to evaluate the predictive performance of our network in interpolating between the diverse conformational states, a third set of MD simulations was performed with the ligand removed from the closed state (closed apo) [20]. We performed two additional repeats of this simulation using the same simulation protocol as Ref. [20]. This produced a total of 1513 conformations representing a transition from the closed-to-open state unseen in either the closed- or open-state simulations.

To test the transfer learning capabilities of the neural network, we selected simulations of p24 and TBE-sE from Ref. [39], HSP from Ref. [20], and SurA from Ref. [40]. For each of these simulation datasets, we created five training sets featuring 1000 conformations and five featuring 100 conformations via random selection.

All our neural networks were trained on the C, C α , N, O (backbone), and C β (side-chain) atoms, as representatives of protein structure. These are sufficient to describe the global conformation (fold) of a protein and the orientation of each side chain. The two extreme conformations in all cases (MurD, p24, TBE-sE, HSP, and SurA) were selected by calculating a RMSD (backbone) matrix considering all the conformations of the training set and picking the pair with the highest RMSD. In all cases, we asked the networks to generate 20 conformations interpolating between these extrema. The RMSD between these extreme conformations, an indicator of protein range of dynamics, was 10.2 Å for MurD, 16.9 Å for p24, 4.9 Å for TBE-sE, 2.8 Å for HSP, and 38.3 Å for SurA.

F. Percentage error calculation

We used the equilibrium values of bonds and angles from the AMBER ff14SB force field [28] to estimate the percentage error in the corresponding bonded parameters modeled by our network. The error over all the bonds and angles present in a conformation was summed to obtain the $\%_{\text{TotalErr}}$ as a measure of accuracy of the network in

generating physically plausible protein structures: $\%_{\text{TotalErr}} = 100(\sum_{\text{bonds}} |r - r_0|/r_0 + \sum_{\text{angles}} |\theta - \theta_0|/\theta_0)$. Similarly, the per-residue errors were calculated as the sum of errors associated with interactions involving any atom within a residue.

G. Analysis of MurD opening angle

The opening angles (azimuth and elevation) of MurD were calculated by aligning the protein along the stable domain (residues 1–298), centering the resulting alignment on the hinge between the mobile and stable domains (center of mass of residues 230–298), and reporting the position of the center of mass of the mobile domain (residues 299–437) in spherical coordinates. A schematic representation of the angles calculated is shown in Fig. 4(a).

Visual inspections of the protein structures and related figures were done with VMD1.9.2 [41] and PyMol [42]. Graphs were plotted using matplotlib [43].

Our neural network is available, under GPLv3 license [44].

ACKNOWLEDGMENTS

The work was supported by the Engineering and Physical Sciences Research Council (EP/P016499/1).

V. K. R., S. C. M., C. G. W., and M. T. D. designed the research, developed the software, carried out the experiments, interpreted the results, and wrote the manuscript.

The authors declare no competing interests.

-
- [1] R. Zwanzig, A. Szabo, and B. Bagchi, *Levinthal's Paradox*, *Proc. Natl. Acad. Sci. U.S.A.* **89**, 20 (1992).
- [2] M. Karplus and J. A. McCammon, *Molecular Dynamics Simulations of Biomolecules*, *Nat. Struct. Mol. Biol.* **9**, 646 (2002).
- [3] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, *arXiv:1312.6114*.
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Nets*, in *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Curran Associates, Inc., 2014), pp. 2672–2680, <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.
- [5] F. Noé, S. Olsson, J. Köhler, and H. Wu, *Boltzmann Generators: Sampling Equilibrium States of Many-Body Systems with Deep Learning*, *Science* **365**, eaaw1147 (2019).
- [6] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, *Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer*, in *Proceedings of the International Conference on Learning Representations, New Orleans, 2019*, <https://openreview.net/forum?id=S1fQSiCcYm>.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, *Deep Learning*, *Nature (London)* **521**, 436 (2015).
- [8] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, *Predicting the Sequence Specificities of DNA- and RNA-Binding Proteins by Deep Learning*, *Nat. Biotechnol.* **33**, 831 (2015).
- [9] J. Zhou and O. G. Troyanskaya, *Predicting Effects of Noncoding Variants with Deep Learning-Based Sequence Model*, *Nat. Methods* **12**, 931 (2015).
- [10] O. Denas and J. Taylor, *Deep Modeling of Gene Expression Regulation in an Erythropoiesis Model*, in *Proceedings of the ICML Workshop on Representation Learning* (Association of Computer Machinery, New York, 2013).
- [11] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, *Improving Computer-Aided Detection Using Convolutional Neural Networks and Random View Aggregation*, *IEEE Trans. Med. Imaging* **35**, 1170 (2016).
- [12] P.-P. Ypsilantis, M. Siddique, H.-M. Sohn, A. Davies, G. Cook, V. Goh, and G. Montana, *Predicting Response to Neoadjuvant Chemotherapy with PET Imaging Using Convolutional Neural Networks*, *PLoS One* **10**, e0137036 (2015).
- [13] T. Zeng, R. Li, R. Mukkamala, J. Ye, and S. Ji, *Deep Convolutional Neural Networks for Annotating Gene Expression Patterns in the Mouse Brain*, *BMC Bioinf.* **16**, 147 (2015).
- [14] B. Zhang, J. Li, and Q. Lü, *Prediction of 8-State Protein Secondary Structures by a Novel Deep Learning Architecture*, *BMC Bioinf.* **19**, 293 (2018).
- [15] S. Long and P. Tian, *Protein Secondary Structure Prediction with Context Convolutional Neural Network*, *RSC Adv.* **9**, 38391 (2019).
- [16] J. Hou, B. Adhikari, and J. Cheng, *DeepSF: Deep Convolutional Neural Network for Mapping Protein Sequences to Folds*, *Bioinformatics* **34**, 1295 (2018).
- [17] R. Zamora-Resendiz and S. Crivelli, *Structural Learning of Proteins Using Graph Convolutional Neural Networks*, <https://doi.org/10.1101/610444>.
- [18] M. Gao, H. Zhou, and J. Skolnick, *DESTINI: A Deep-Learning Approach to Contact-Driven Protein Structure Prediction*, *Sci. Rep.* **9**, 3514 (2019).
- [19] A. W. Senior, R. Evans, J. Jumper *et al.*, *Improved Protein Structure Prediction Using Potentials from Deep Learning*, *Nature (London)* **577**, 706 (2020).
- [20] M. T. Degiacomi, *Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space*, *Structure* **27**, 1034 (2019).
- [21] J. A. Bertrand, G. Auger, E. Fanchon, L. Martin, D. Blanot, J. van Heijenoort, and O. Dideberg, *Crystal Structure of UDP-N-Acetylmuramoyl-L-Alanine: D-Glutamate Ligase from Escherichia Coli*, *EMBO J.* **16**, 3416 (1997).
- [22] J. A. Bertrand, E. Fanchon, L. Martin, L. Chantalat, G. Auger, D. Blanot, J. van Heijenoort, and O. Dideberg, *“Open” Structures of MurD: Domain Movements and Structural Similarities with Folylpolyglutamate Synthetase*, *J. Mol. Biol.* **301**, 1257 (2000).
- [23] J. A. Bertrand, G. Auger, L. Martin, E. Fanchon, D. Blanot, D. Le Beller, J. van Heijenoort, and O. Dideberg, *Determination of the MurD Mechanism through Crystallographic Analysis of Enzyme Complexes*, *J. Mol. Biol.* **289**, 579 (1999).

- [24] R. Šink, M. Kotnik, A. Zega, H. Barreateau, S. Gobec, D. Blanot, A. Dessen, and C. Contreras-Martel, *Crystallographic Study of Peptidoglycan Biosynthesis Enzyme MurD: Domain Movement Revisited*, *PLoS One* **11**, e0152075 (2016).
- [25] R. Šink, H. Barreateau, D. Patin, D. Mengin-Lecreulx, S. Gobec, and Didier Blanot, *MurD Enzymes: Some Recent Developments*, *Biomol. Concepts* **4**, 539 (2013).
- [26] T. M. Belete, *Novel Targets to Develop New Antibacterial Agents and Novel Alternatives to Antibacterial Agents*, *Hum. Microb. J.* **11**, 100052 (2019).
- [27] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevX.11.011052> for additional benchmarks on neural network performance and characterization of associated latent spaces.
- [28] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, and C. Simmerling, *ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB*, *J. Chem. Theory Comput.* **11**, 3696 (2015).
- [29] M.-y. Shen and A. Sali, *Statistical Potential for Assessment and Prediction of Protein Structures*, *Protein Sci.* **15**, 2507 (2006).
- [30] S. Monaco-Malbet, C. Berthet-Colominas, A. Novelli, N. Battaï, N. Piga, V. Cheynet, F. Mallet, and S. Cusack, *Mutual Conformational Adaptations in Antigen and Antibody upon Complex Formation between an Fab and HIV-1 Capsid Protein p24*, *Structure* **8**, 1069 (2000).
- [31] F. A. Rey, F. X. Heinz, C. Mandl, C. Kunz, and S. C. Harrison, *The Envelope Glycoprotein from Tick-Borne Encephalitis Virus at 2 Å Resolution*, *Nature (London)* **375**, 291 (1995).
- [32] C. Bagn eris, O. A. Bateman, C. E. Naylor, N. Cronin, W. C. Boelens, N. H. Keep, and C. Slingsby, *Crystal Structures of α -Crystallin Domain Dimers of α B-Crystallin and Hsp20*, *J. Mol. Biol.* **392**, 1242 (2009).
- [33] E. Bitto and D. B. McKay, *Crystallographic Structure of SurA, a Molecular Chaperone that Facilitates Folding of Outer Membrane Porins*, *Structure* **10**, 1489 (2002).
- [34] Z. Zhang, S. Witham, and E. Alexov, *On the Role of Electrostatics in Protein–Protein Interactions*, *Phys. Biol.* **8**, 035001 (2011).
- [35] J. B. Adams, *Bonding Energy Models*, in *Encyclopedia of Materials: Science and Technology*, edited by K. H. J rgen Buschow, Robert W. Cahn, Merton C. Flemings, Bernhard Ilshner, Edward J. Kramer, Subhash Mahajan, and Patrick Veyssi re (Elsevier, Oxford, 2001), pp. 763–767, <https://doi.org/10.1016/B0-08-043152-6/00146-7>.
- [36] A. R. Leach, *Molecular Modelling: Principles and Applications* (Pearson Education, 2001).
- [37] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*, [arXiv:1511.07289](https://arxiv.org/abs/1511.07289).
- [38] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [39] M. T. Degiacomi and M. Dal Peraro, *Macromolecular Symmetric Assembly Prediction Using Swarm Intelligence Dynamic Modeling*, *Structure* **21**, 1097 (2013).
- [40] A. N. Calabrese, B. Schiffrin, M. Watson, T. K. Karamanos, M. Walko, J. R. Humes, J. E. Horne, P. White, A. J. Wilson, A. C. Kalli, R. Tuma, A. E. Ashcroft, D. J. Brockwell, and S. E. Radford, *Inter-Domain Dynamics in the Chaperone SurA and Multi-Site Binding to Its Outer Membrane Protein Clients*, *Nat. Commun.* **11**, 2155 (2020).
- [41] W. Humphrey, A. Dalke, and K. Schulten, *VMD: Visual Molecular Dynamics*, *J. Mol. Graphics* **14**, 33 (1996).
- [42] The PyMol Molecular Graphics System, Version 1.8, Schr dinger, LLC, 2015.
- [43] J. D. Hunter, *Matplotlib: A 2D Graphics Environment*, *Comput. Sci. Eng.* **9**, 90 (2007).
- [44] <https://github.com/degiacom/molearn>