DiffNets: deep learning the structural determinants of proteins biochemical properties by comparing different structural ensembles

Michael D. Ward[1,3], Maxwell I. Zimmerman[1,3], Artur Meller[1,3], Moses Chung[1,3], S.J. Swamidass[2], and Gregory R. Bowman[1,3]*

[1]Department of Biochemistry & Molecular Biophysics, Washington University School of Medicine, St. Louis, MO 63110, USA
[2]Department of Pathology & Immunology, Washington University School of Medicine, St. Louis, MO 63110, USA
[3]Center for the Science and Engineering of Living Systems, Washington University in St. Louis, St. Louis, MO 63110, USA

*To whom correspondence should be sent: g.bowman@wustl.edu

Abstract
Understanding the structural determinants of a protein's biochemical properties, such as activity and stability, is a major challenge in biology and medicine. Comparing computer simulations of protein variants with different biochemical properties is an increasingly powerful means to drive progress. However, success often hinges on dimensionality reduction algorithms for simplifying the complex ensemble of structures each variant adopts. Unfortunately, common algorithms rely on potentially misleading assumptions about what structural features are important, such as emphasizing larger geometric changes over smaller ones. Here we present DiffNets, self-supervised autoencoders that avoid such assumptions, and automatically identify the relevant features, by requiring that the low-dimensional representations they learn are sufficient to predict the biochemical differences between protein variants. For example, DiffNets automatically identify subtle structural signatures that predict the relative stabilities of β-lactamase variants and duty ratios of myosin isoforms. DiffNets should also be applicable to understanding other perturbations, such as ligand binding.

A mechanistic understanding of how a protein's sequence determines its structural preferences and, ultimately, its biochemical properties is crucial for advancing our understanding of fundamental biology and for applications in precision medicine and protein engineering. Sequence variations can modulate a protein's biochemical properties in a deleterious manner leading to morbidity and mortality,[1,2] or in a manner that can improve a species fitness, e.g. conferring the ability to metabolize new substrates.[3] Moreover, entire protein families with a wide range of functions and biochemical properties emerge after long timescale evolution of protein sequences. In either case, identifying the structural and dynamical differences between protein variants is a powerful means to understand the mechanism that connects a protein's sequence and biochemical properties.[4–9] Streamlining this process would make it easier to infer the behavior of new protein variants, which would accelerate protein engineering and the interpretation of newly discovered variants. Understanding the structural basis for protein function and dysfunction can also accelerate the development of drugs and other therapeutics.

Identifying the structural features that determine the biochemical differences between protein variants is often a difficult challenge, requiring one to consider the entire ensemble of structures that a protein adopts. Techniques like crystallography and cryoEM sometimes reveal dramatic structural differences between protein variants that readily explain their biochemical differences. However, there are also many cases where structural snapshots do not provide a clear explanation for the differences between variants,[10] suggesting that one must consider the entire ensemble of thermally accessible configurations these proteins adopt to understand the biochemical differences between them[11–13]. Molecular dynamics simulations can provide access to these ensembles.[14] However, there are many factors that make comparing these ensembles difficult. First of all, proteins have thousands of degrees of freedom that enable them to adopt an enormous number of different configurations[15,16]. Moreover, two ensembles may be highly overlapping, requiring one to identify differences in the probabilities of structural features that are present in both ensembles, rather than simply identifying features that are only present in one ensemble. For example, mutations in the enzyme TEM β-lactamase were found to determine its specificity by modulating the relative probabilities of different structures,[12] but all the variants considered had a reasonable probability of adopting any of these structures.

Dimensionality reduction algorithms play a crucial role in dealing with the enormity of conformational ensembles. Many powerful algorithms have been developed and employed successfully, but the utility of each is limited by assumptions that are not universally appropriate. For example, principal component analysis (PCA)[17,18] finds linear combinations of features that retain as much of the geometric variance in the original data as possible, effectively assuming that large structural changes are more important than subtle ones. Unfortunately, there are many cases where this assumption is invalid, as in enzymes where arbitrary motions of a large floppy loop may dwarf subtle but functionally-relevant sidechain motions in the active site. Autoencoders[19] are a more powerful alternative since they consider nonlinear combinations of features. These neural networks learn a low-dimensional projection of data—called the latent space—that is optimized to produce a high-fidelity geometric reconstruction of a protein configuration (Fig. 1). However, like PCA, autoencoders still focus on capturing large geometric variations. Time-lagged independent component analysis (tICA)[20,21] is another common approach. It is similar to PCA but focuses on slowly varying degrees of freedom rather than emphasizing large geometric changes. However, there are many situations where the conformational changes of interest are fast relative to others (e.g. allostery within the native ensemble that is faster than folding and unfolding of the protein). Another recent approach, VAMPnets,[22] combines ideas from autoencoders and tICA to achieve a dimensionality reduction that maps protein structures to metastable states. This allows VAMPnets to capture non-

linearities that tICA cannot, but the assumption that slowly varying degrees of freedom are more important than faster ones is still limiting in many cases.

We hypothesized that requiring a dimensionality reduction algorithm to predict the biochemical differences between protein variants would be a powerful means to ensure that it identifies the relevant structural differences without being misled by a priori assumptions. Instead of assuming what type of variation is important (e.g. that large structural changes are more important than smaller ones), such an algorithm would simply assume there are differences between two or more classes of data and then search for features that separate these classes.

To test this hypothesis, we introduce DiffNets, a dimensionality reduction algorithm that uses a self-supervised autoencoder to learn features of a protein's structural ensemble that are predictive of the biochemical differences between protein variants (Fig. 1). While we focus on protein variants, the algorithm should be equally applicable to other perturbations, such as understanding the impact of post-translational modifications and interactions with binding partners. DiffNets takes two inputs: 1) a set of molecular dynamics simulations for each protein variant and 2) the biochemical property of interest (e.g. stability or activity) for each variant. The algorithm then learns a low-dimensional projection (latent space) of the protein structures that is explicitly organized to separate structural configurations based on how closely they are associated with the biochemical property of interest. DiffNets achieve this by combining supervised autoencoders[23] with self-supervision. Supervised autoencoders are multi-task networks. Like standard (unsupervised) autoencoders, they must learn a low-dimensional projection of the data that retains sufficient geometric information to reconstruct the original high-dimensional input (Fig. 1, left). However, supervised autoencoders add the additional requirement that the low-dimensional projection of the data be sufficient to predict a label, in this case one related to the biochemical property of interest. This second requirement forces the dimensionality reduction to dedicate representational power to identifying degrees of freedom that are important for the label instead of focusing exclusively on large structural changes. The classification task can be based on the entire latent space to minimize assumptions, or on a subset of the inputs (e.g. the region around a mutation, as in Fig. 1) to focus attention on critical areas. Self-supervision provides an automated way to deal with the fact that we know the biochemical properties of variants (i.e. their entire structural ensemble), but the association between any specific structure and that biochemical property is unknown. This problem is non-trivial because there is likely to be overlap between ensembles (i.e. structures that are visited by all variants). Therefore, classifying all structures from variants without the property of interest as different than all structures from variants with the property is likely a misleading oversimplification. To overcome this limitation, we present an expectation maximization scheme that iteratively updates training labels to identify a subset of structures that are more probable for variants with the biochemical property of interest while allowing for overlap between the conformational ensembles of different variants.

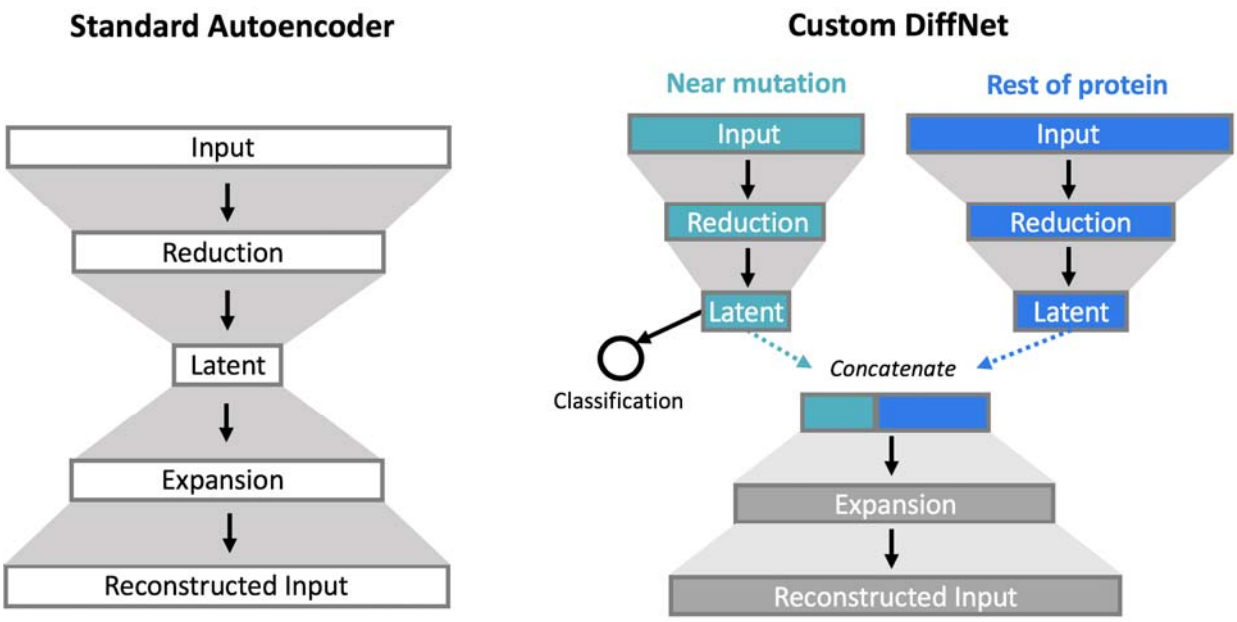


Figure 1. Standard autoencoder architecture (left) and an example DiffNet architecture. Autoencoders have an encoder that compress the input data to a bottleneck, or latent, layer and a decoder that expands the latent representation to reconstruct the original input. The DiffNet adds a classification task to the latent space. In the example shown, the input is split into two encoders. One is a supervised encoder that operates on atoms near the mutation and must predict the biochemical property associated with a structure. The second encoder is unsupervised and operates on the rest of the protein. The latent layers from these two encoders are concatenated and trained to reconstruct the original input.

To test the performance of DiffNets, we apply them to a set of four TEM β-lactamase variants, which differ by single point mutations, and to a set of eight myosin isoforms. First, we demonstrate how the DiffNet classification task alters dimensionality reduction of protein structures compared to standard autoencoders. Then, we use DiffNets to recapitulate known differences in β-lactamase variants' folded ensembles that are predictive of changes in stability between variants. The relevant changes are geometrically subtle (< 1 Å distance change) compared to other motions, and thus, it originally took our group several months to identify them. Therefore, attempting to recapitulate this result is a challenging test case for new methods, such as DiffNets. Finally, we use DiffNets to understand the structural determinants of duty ratio (i.e. the amount of time a myosin protein spends attached to actin) among eight myosin isoforms. This is a difficult test case since small loop motions are critical for determining duty ratio, which is difficult to pick out in large (e.g. ~800 residues) myosin motor proteins. Further, the underlying amino acid sequences of isoforms are highly divergent, so success on this task would demonstrate that DiffNets are applicable to variants with more complex perturbations compared to single-point mutations.

Results and Discussion

The DiffNet architecture.

The DiffNet architecture is based on an autoencoder, which is a deep learning framework commonly used for dimensionality reduction[4,24,33–35,25–32](Fig. 1). Like standard autoencoders, DiffNets connect an encoder and decoder network to compress and reconstruct input data, respectively. In our case, the input is protein XYZ coordinates (C,CA,N,CB) from a simulation frame, which are whitened for normalization (see methods). First, the encoder network transforms the input to progressively reduce the dimensionality of the input to a bottleneck layer, called the latent space. Then, the latent space vector is used as input to the decoder network that attempts to reconstruct the original input. Mechanically, both the encoder and decoder operate via successive matrix multiplications and non-linear activation functions. DiffNets (and autoencoders) are initialized with random matrix multiplications, and the network improves by iteratively tuning the matrix values (weights) by training across many examples. Concretely, the weights are trained to minimize a loss function that measures the difference between the input and output of the model, called the input reconstruction error. Ultimately, if a DiffNet (or autoencoder) can compress and then reconstruct the original input with high accuracy, this implies that the low-dimensional latent space vector retains the salient features that describe the input.

Inspired by supervised autoencoders[23], DiffNets augment autoencoders with a loss function that measures how accurately the latent space vector performs a user-defined classification task (e.g. did the protein structure come from a wild-type or variant simulation?). Therefore, DiffNets must learn weights that simultaneously minimize protein reconstruction error and classification error. The constraint to minimize protein reconstruction error enforces that the low-dimensional representation of data retains a structural basis, and the classification constraint is designed to reconfigure the latent space such that data points are separated to highlight differences between datasets (e.g. biochemical differences between protein variants). While supervised autoencoders have been previously used as a way to obtain better performance on a classification task[23], we use the classification task to learn a more interpretable low-dimensional projection of data. Additionally, we propose an expectation maximization scheme such that classification labels are updated between DiffNet training epochs. This self-supervision provides an automated way to deal with the fact that we know the biochemical properties of variants (i.e. their entire structural ensemble), but the association between any specific structure and that biochemical property is unknown.

The DiffNet architecture can be split to focus the classification task on a region of interest within a protein. If there is a region of interest known *a priori* (e.g. region around a mutation, or an enzyme active site) the input may be split into two encoder networks. In this case, only the encoder with inputs from the region of interest performs a classification task, then the latent spaces from each encoder are concatenated for input to the decoder (see Fig. 1). This split architecture guides a DiffNet to search in the region of interest to find differences between variants. This is a reasonable default to use when studying single point mutations as the region of a mutation is root of differences between variants. Moreover, classifying based on a region of interest does not preclude the identification of relevant distal structural differences between variants. If a mutation causes biochemically relevant differences at distal regions then these regions are inherently linked to the state of the region of the mutation and, thus, are implicitly linked to the classification task.

The classification task reorganizes the latent space to emphasize biochemically-important structural features.

Dimensionality reduction algorithms are only helpful for identifying differences between two classes of data if the two classes of data are separated in the latent space. Unsupervised autoencoders learn a latent representation of data that focuses on large geometric variations, so structures with large geometric differences are separated, while structures with subtle differences are close together. As a result, if biochemical differences between protein variants are related to subtle geometric changes, then the variants will be highly overlapping in the latent space and thus, the autoencoder will fail to provide a useful way to distinguish variants. We hypothesized that augmenting a standard autoencoder with a classification task, as with DiffNets, would reorganize the latent space to highlight relevant differences between datasets, even if they are subtle structurally.

In order to test this hypothesis, we applied DiffNets and autoencoders to a set of variants of the enzyme TEM β-lactamase. β-lactamase is an enzyme that confers bacteria with antibiotic resistance by metabolizing β-lactam drugs like penicillin[36]. Bacteria are quick to evolve new variants of TEM that have activity against new drugs, but these mutations are often destabilizing, so compensatory mutations evolve to restore stability[37–39]. M182T is one stabilizing mutation that frequently appears in clinical isolates[40,41]. While crystal structures of the wild-type and M182T proteins had been solved, comparing them did not provide a conclusive mechanism for stabilization capable of predicting the impact of other variants. Recently, our group combined simulations, NMR experiments, and x-ray crystallography to demonstrate that compaction of helix 9 is a structural signature that distinguishes more stable variants (like M182T) from less stable ones (Fig. 2). This compaction is associated with stronger h-bonds along helix 9 that stabilize this secondary structure element. Helix 9 is part of a crucial interdomain interface, so stabilizing it ultimately stabilizes the native state relative to an intermediate where one domain is at least partially unfolded. Importantly, this helix compaction includes distance changes of less than 1 Ångstrom between hydrogen bonding partners. Given that this is geometrically subtle compared to nearby loop motions, we expect that compact and extended helix states will not be well separated in the latent space of a standard autoencoder. However, we do expect that a DiffNet trained to classify compact and extended helix states will learn a latent space that separates these states.

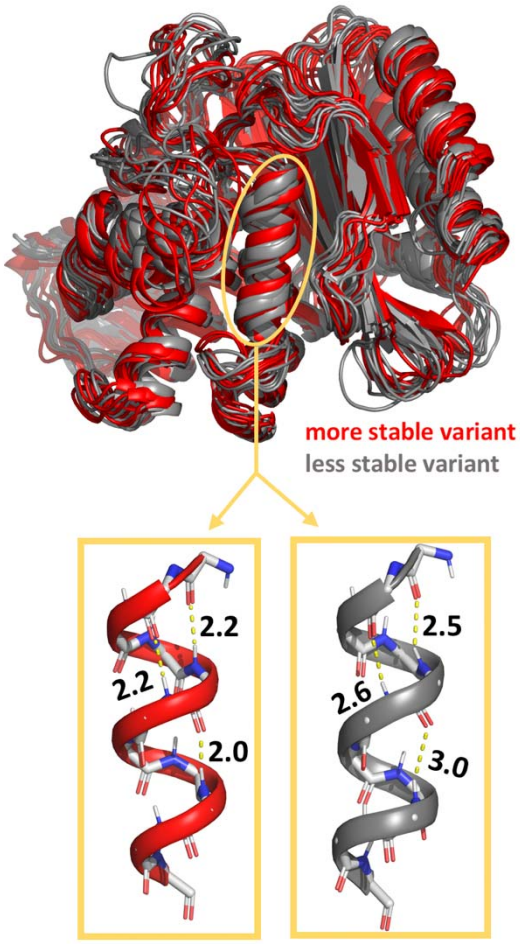


Figure 2. Structural configurations sampled from molecular dynamics simulations of wild-type TEM β-lactamase (grey) and an M182T variant (red) that is far more stable. Helix 9 is circled in yellow and shown below in a compact configuration (left, red) and a more extended configuration (right, grey). Hydrogen bond distances are shown in Ångstroms.

To evaluate if the DiffNet classification layer alters the latent space in a way that helps identify differences between two classes of data, we compared the latent space of DiffNets to the latent space of unsupervised autoencoders after training on a dataset that includes two classes of data distinguishable by a subtle difference in helix 9 compaction. First, we curated a dataset of ~180,000 simulation frames from wild-type and M182T simulations where half of the frames have a compact helix 9 (helix compaction criteria described in Methods) and half have a more extended helix. Then, we trained DiffNets and unsupervised autoencoders using a split architecture described in the methods and visualized in Figure 1b. The DiffNets and autoencoders we trained were identical, except the DiffNet has an additional output layer such that it has to classify helix 9 as compact or extended in addition to reconstructing protein structures. The classification labels are not updated with expectation maximization in this case. This dataset was selected specifically to evaluate how the classification task of the DiffNet alters the dimensionality reduction compared to a standard autoencoder. In a normal setting we would not have *a priori* knowledge about the importance of helix 9 compaction. However, this is an important test to determine if adding a classification task can reorganize the latent space to highlight differences between datasets, which is a property that DiffNets will ultimately need to identify differences between variants.

Requiring DiffNets to perform a classification task in tandem with dimensionality reduction successfully reconfigures the latent space to disentangle compact helix configurations from more extended helix configurations. First, we note that DiffNets and unsupervised autoencoders have similar ability to reconstruct protein structures (~1 Ångstrom error - see Figure 3) using as few as three latent variables and as many as fifty, which is in line with another study reporting autoencoder reconstruction error[34]. To compare latent spaces, we analyze a split architecture that has twenty-five latent variables including three in encoder A (which receives input including helix 9 and performs the classification task in the DiffNet) and twenty-two in encoder B (takes input from the rest of the protein). This architecture provides a low reconstruction error (< 1 Ångstrom) and few enough latent variables so that all dimensions in encoder A's latent space can be visualized. In the unsupervised autoencoder, simulation frames of compact and extended helices are overlapping in encoder A's latent space (Fig. 4a). This demonstrates that training an unsupervised autoencoder on two classes of data does not necessarily yield a latent representation that provides any insight into how the two classes of data are different. To explore this point further, we held the autoencoder's latent space constant and then trained it to classify whether a structure has a compact or extended helix 9 (i.e. performed logistic regression). The resulting receiver operating characteristic (ROC) curve, which measures classification performance, shows a classification performance similar to random guessing (area under the curve [AUC]=0.54) providing further evidence that the latent representation does not help distinguish the two classes of data. In contrast, the DiffNet encoder A latent space clearly separates the two classes of data (Fig. 4a) and has excellent performance classifying compact and extended helix states (AUC=0.91, Fig. 4b). This result demonstrates that adding a classification component to the learning task provides a powerful means to learn a low dimensional representation that highlights crucial differences between datasets. It follows that DiffNets trained with a classification task that must predict a biochemical property should learn a low dimensional representation of data that highlights structural features that are predictive of biochemical differences between protein variants.

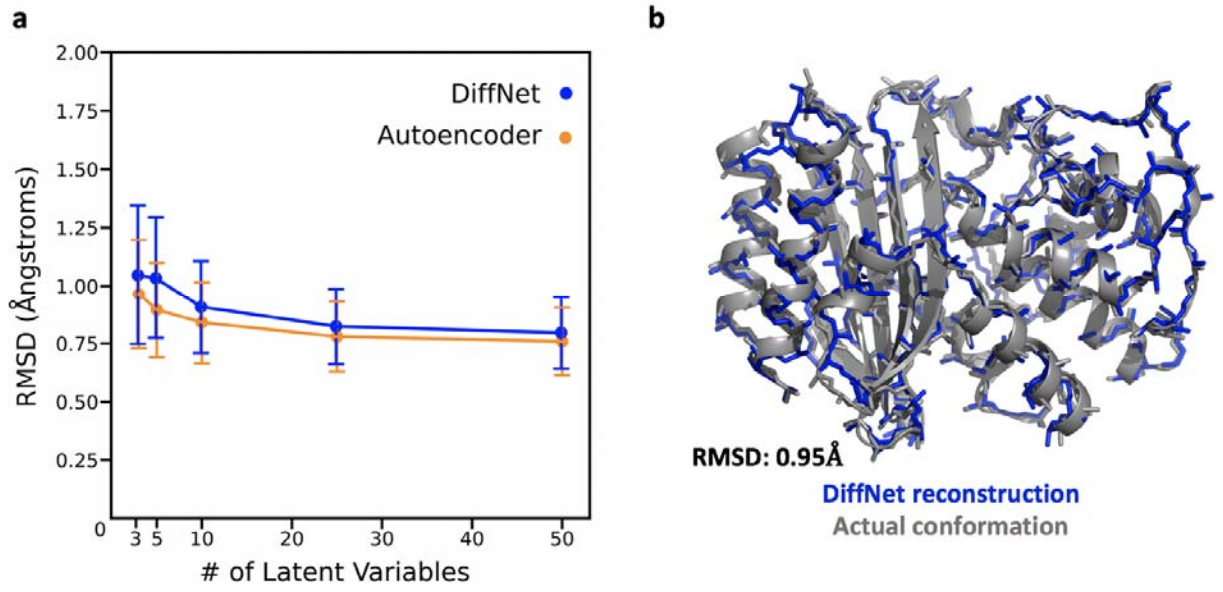
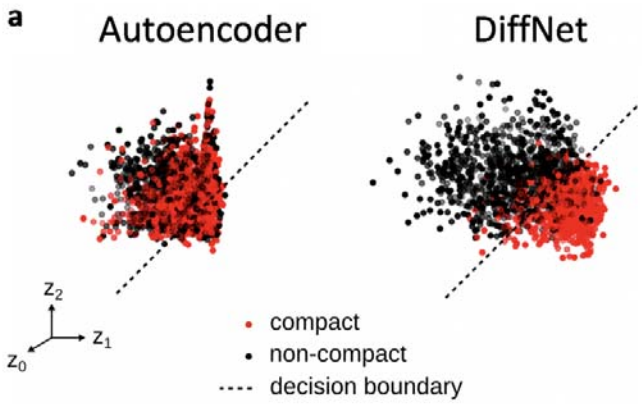


Figure 3. Autoencoders and DiffNets can both compress protein structures and then reconstruct them. **a** Reconstruction error plots showing the RMSD between a protein structure from simulation and the corresponding protein structure generated by unsupervised autoencoders (yellow) or DiffNets (blue). **b** structure representing the difference between a structure generated by the DiffNet (blue) vs. the actual conformation from simulation (grey) when training with 3 latent variables.



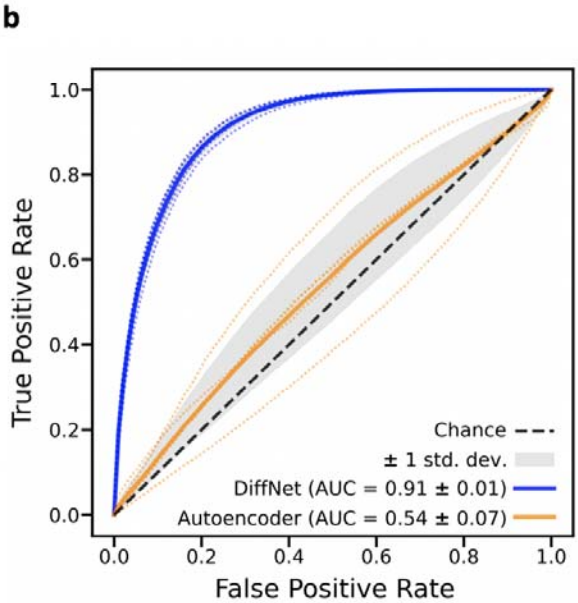


Figure 4. Adding a classification component to the learning task (as in DiffNets) results in a latent representation that separates different datasets more clearly than an unsupervised autoencoder. **a** Simulation frames that have a compact helix (red) and an extended helix (black) are projected onto the three-dimensional latent space learned by an unsupervised autoencoder (left) and a DiffNet (right). The decision boundary (black dotted line) indicates the plane that each neural network uses to separate compact helix states from extended helix states. **b** ROC curve showing the average classification performance of the unsupervised autoencoder (dark yellow) and the DiffNet (dark blue) as well as the performance for each of the 5 folds of cross validation (faded dotted lines).

Self-supervised DiffNets learn structural signatures associated with protein stability.

While the classification task can help DiffNets learn a useful dimensionality reduction, realizing this potential is non-trivial because we know the biochemical properties of variants (e.g. their entire structural ensembles) but not individual structures. The simplest approach to providing these classification labels would be to assign ones to structures from simulations of variants with the biochemical property of interest and zeros to structures from simulations of variants without the property. However, it is likely that variants fall on a continuum rather than having a biochemical property or not, that their conformational ensembles overlap, and that only a subset of conformations are relevant for determining the property of interest.

This problem is similar to multiple instance learning. During multiple-instance learning, learners are given bags of training examples where each bag is labelled negative, indicating that the bag contains all negative examples, or positive, indicating that there are at least some positive examples in the bag. The learner then must figure out how to label all of the individual instances as positive or negative by identifying features that are consistent in positive bags, but absent in negative bags. This is similar to our situation where we know the biochemical property of each protein variant (i.e. negative bag or positive bag), but we do not know if a given structural configuration is associated with a biochemical property, or inconsistent with a biochemical property.

We propose a self-supervised approach for learning the relationship between individual structures and the biochemical property of interest using an iterative expectation maximization algorithm based on work from Zaretski et. al.[42] Expectation maximization is a statistical method that allows the parameters of a model to be fit, even when the outputs of the model cannot be observed directly in the training data[43] (i.e. when they are hidden). In our case, the hidden variables are labels for each structure that specify the probability that a structure is associated with the biochemical property of interest. These labels are initially set to ones for all structures from variants with a given biochemical property (e.g. more stable β-lactamase variants) and zeros for variants without that property (e.g. variants with lower stabilities). Then the expectation maximization algorithm iteratively alternates between a maximization step and an expectation step to identify a self-consistent set of labels. During the maximization step, a DiffNet is trained to predict the current labels for each structure. Then, the expectation step refines the training labels by computing the expected values of the labels, y, using the output from the DiffNet, $\hat{y}$, conditioned on constraints about what fraction of structures from each variant we expect to be associated with the property of interest. This constraint provides a way to enforce that more high probability values are assigned to structures from variants with the biochemical property. The expectation is the probability-weighted average of all binary realizations of binomial distributions parameterized by $\hat{y}$, excluding binary realizations that do not meet the constraint

$$y_i = E[\,\hat{y}_i \mid S_L \leq \hat{y}_r \leq S_U\,]$$

$$= P(\hat{y}_i \text{ is } 1) * \left(\frac{P(S_L - 1 \leq \hat{y}_r - \hat{y}_i \leq S_U - 1}{P(S_L \leq \hat{y}_r \leq S_U)}\right)$$

where y_i is the current label for each individual frame, $\hat{y}_i$ is the DiffNet output, S_L and S_U are the lower and upper bounds on how many conformations in a batch are associated with the biochemical property, $\hat{y}_r$ is the sum of the binary outcomes of a batch which contains conformation i, $P(S_L - 1 \leq \hat{y}_r - \hat{y}_i \leq S_U - 1)$ is the probability that the number of conformations

in a batch is within the limits if conformation i is ignored, and $P(S_L \leq \hat{y}_r \leq S_U)$ is the probability that the number of conformations in a batch is within the limits, including conformation i. Ultimately, the desired outcome is that the expectation maximization algorithm redistributes training labels from all 0s and 1s for simulation frames of variants without and with a biochemical property, respectively, to values that indicate the probability that a given structural configuration is associated with the biochemical property of interest. This mechanism is self-supervised since the training labels are learned by the algorithm, rather than explicitly curated.

To test this approach, we trained a self-supervised DiffNet to identify structural preferences that distinguish two highly stable β-lactamase variants (M182T and M182S) from two less stable variants (wild-type [WT] and M182V). In this case, the DiffNet receives no *a priori* information about features, like helix 9 compaction, that are associated with increased stability in M182T and M182S. If self-supervision of DiffNets works as expected, then training should produce a latent space where it is easy to identify the structural features that are associated with the stability of M182T and M182S, relative to WT and M182V. For example, we expect to see structural configurations with a compact helix 9 in one region of the latent space and structures with a more extended helix elsewhere. Beyond helix compaction, DiffNets may even capture additional structural features that were missed in our previous manual analysis. To evaluate if the DiffNet learns these biochemically relevant structural differences between variants, we trained a DiffNet on 6.5μs of simulation data for each variant: M182T, M182S, WT, and M182V. All frames from M182T and M182S (highly stable variants) were initially assigned classification labels of 1, and simulation frames from M182V and WT were initially assigned 0s. During the EM procedure, we calculate the expected values (updated labels) conditioned on the constraint that 0-30% of less stable variants frame are likely to be stabilizing, and 60-90% of frames for highly stable variants.

Using expectation maximization, the DiffNet learns a low-dimensional representation that accurately identifies that helix 9 compaction is associated with highly stable variants. Specifically, as we scan along structures with DiffNet output labels ranging from zero to one (i.e. structures associated with low stability to high stability), we find that three key hydrogen-bonding distances at the end of helix 9 decrease (Fig. 5a). As explained previously, we expect there to be overlap between the structural ensembles of highly stable and less stable β-lactamase variants. Indeed, the labels learned by the DiffNet are not perfectly correlated with helix 9 compaction (AUC=0.73) (Fig. 5b). We are encouraged by this AUC score because it indicates that the DiffNet output label is strongly influenced by helix compaction, but must also be a function of other structural features, as expected.

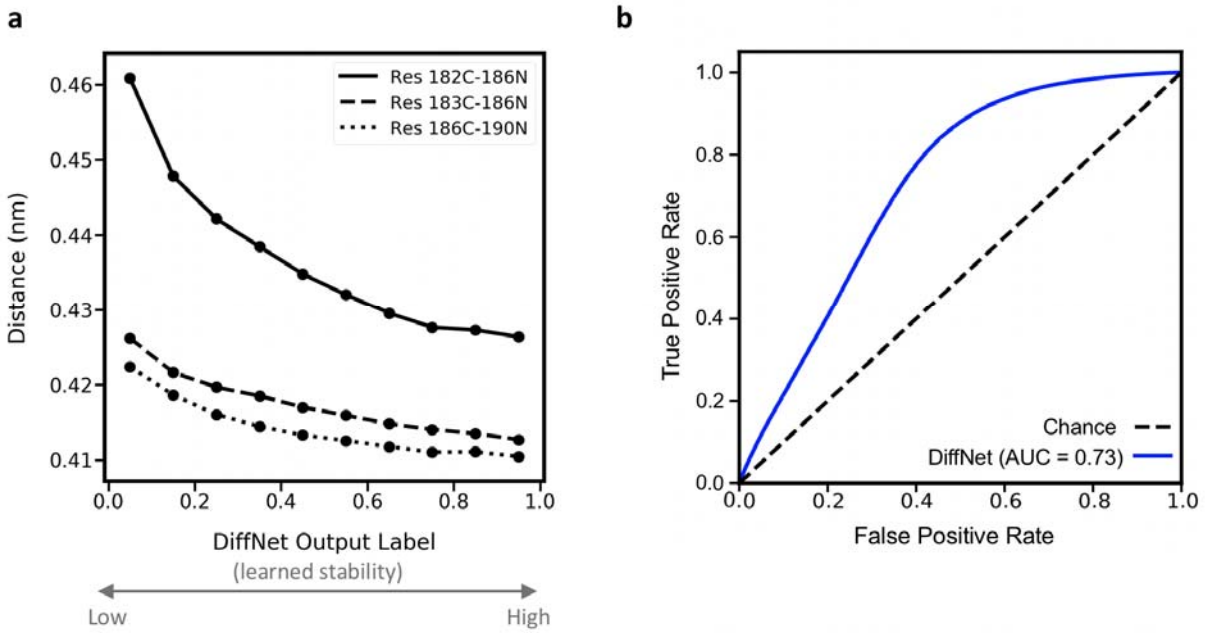


Figure 5. Self-supervised DiffNets identify the importance of helix 9 compaction without any a priori knowledge of the structural features associated with stability. a Three key hydrogen bond lengths in helix 9 as a function of the DiffNets label, which ranges from zero for structures associated with low stability to one for structures associated with high stability. The distances are between the carbonyl carbon of the i'th residue and the nitrogen of the (i+4)'th residue. Standard error bars are not visible since the standard error is smaller than scatter points. **B** ROC curve showing the performance of the DiffNet when classifying compact vs. extended helix 9 states, which the DiffNet was not explicitly trained on.

While many deep learning approaches are criticized for their lack of interpretability, the DiffNet architecture provides opportunities to understand what the network learned, which provides biophysical insight. To automate DiffNet interpretation, we measured all inter-atom distances within 1nm of the mutation using 2000 cluster centers calculated from all simulations and then measured the linear correlation between each distance and the DiffNet output label. We plot the top 1% of distances correlated with the DiffNet output label to visualize the conformational changes that the DiffNet views as important for distinguishing stable variants from less stable variants. Encouragingly, the distance correlations strongly point to helix 9 compaction as an important feature of more stable variants (Fig. 6). While the helix compaction is striking, DiffNets also captured other trends that our previous computational analyses did not detect. For example, our NMR data suggested that the packing between helix 9 and adjacent β-sheet differs in more stable vs less stable variants,[5] but our computational analysis did not detect a clear trend. On the same simulation dataset, the DiffNet clearly learns that this interface becomes more tightly packed for more stable variants (Fig. 6). Moreover, the distance correlations line up with NMR data perfectly in many instances. For example, the distance correlations show a compaction between residue 189, in the middle of helix 9, and residue 263 on the β-sheet, which both have chemical shift perturbations for M182T and M182S (more stable variants) relative to the wild-type. Often times the important features that distinguish protein variants can be complicated and, therefore, easily missed even with months of analysis. DiffNets can learn complicated features and help automate the process of identifying biochemically relevant structural features that distinguish protein variants.

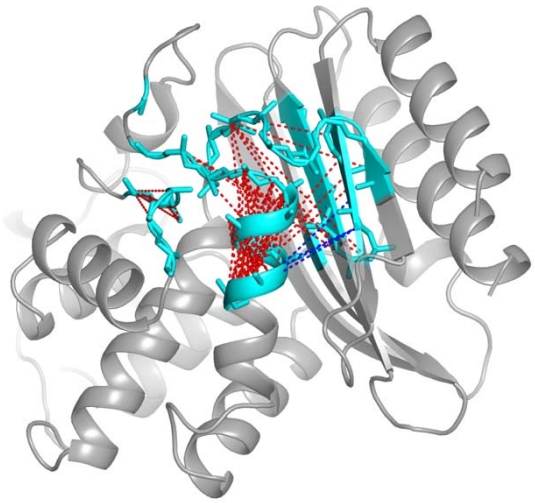


Figure 6. Visualization of the features that DiffNets find important for increased stability of M182T and M182S variants. Protein atoms are colored cyan if they are near the mutation, which indicates that they were included in the classification task and considered for the distance correlation calculation. Dotted lines indicate distances between two atoms that change in a way that is strongly correlated with an increased DiffNet output label. Red indicates the atoms move closer together as the output label increases, blue indicates atoms moving away from each other.

DiffNets predictive power extends to variants not seen during training.

To further test if the DiffNet learned structural features predictive of stability we used the DiffNet to predict if the stability of M182N (not seen during training) is more similar to wild-type and M182V (less stable variants) or M182T and M182S (more stable variants). Since the network never saw any M182N data during training, success would suggest that the network has really learned a useful set of features, rather than simply memorizing which structures came from highly stable variants and which from less stable variants. The M182N variant is also a challenging test case because biochemical intuition suggests that it should be capable of capping helix 9 and stabilizing β-lactamase in a similar manner to other mutations, such as M182T and M182S. However, our past modeling efforts predicted that M182N has the same stability as wild-type because of a competition between two alternative hydrogen-bonding networks, a minor population that resembles highly stable variants and a dominant population that is more similar to less stable variants. Importantly, our subsequent experiments confirmed this prediction[5]. To test if our DiffNet can recapitulate this result, we used it to classify 6.5μs of simulation data from the M182N variant and assessed the distribution of labels predicted by the DiffNet.

The DiffNet accurately groups M182N with the less stable variants, suggesting it has learned generalizable features related to protein stability. The DiffNet's output labels for variants that were in the training set are as expected; more stable variants are grouped together at high values, and less stable variants group together at lower values (Fig. 7). Interestingly, there is some overlap between the more stable and less stable variants, which suggests that self-supervision helped the DiffNet learn that there is overlap between the structural ensembles of the different variants, and not every structure is clearly associated with higher or lower stability. Importantly, the variant not seen during training, M182N, has a major peak around zero, indicating a correct classification with the less stable variants (Fig. 7). In agreement with our

past analysis, M182N also has a minor population that more closely resembles the highly stable variants. This success indicates that DiffNets could be used to predict the biochemical properties of novel variants that were not seen during training. Of course, the success of such predictions will likely depend on the conformational space of the novel variant having high overlap with variants that were used to train DiffNets.

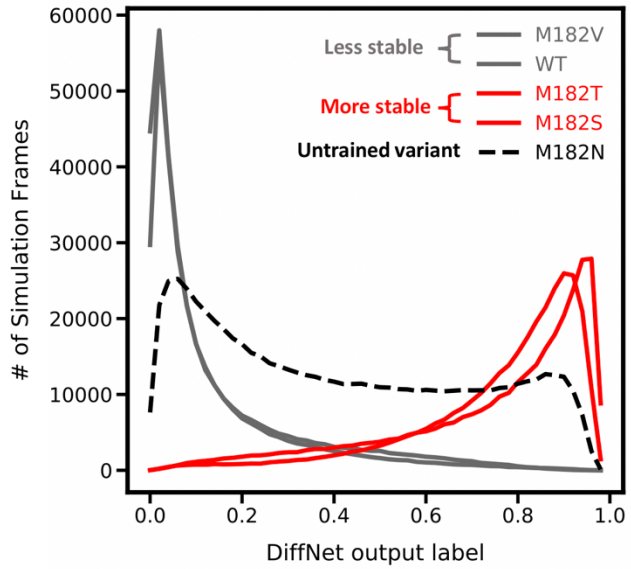


Figure 7. DiffNet groups M182N with less stable variants. Histogram shows the DiffNet output label across all simulation frames from M182Tand M182S (red – highly stable variants in training set), WT and M182V (grey – less stable variants in training set), and M182N (dotted black – not in training set).

DiffNets works for other proteins and more divergent sequences.

In order to explore the broad applicability of DiffNets, we also trained a self-supervised DiffNet to identify structural features that distinguish high duty myosin motor domains from low duty myosins. Myosins are a ubiquitous class of motor proteins that perform an extraordinary diversity of functions despite sharing a common mechanochemical cycle.[44] In order to perform roles as diverse as muscle contraction and intracellular transport, myosins have precisely tuned their duty ratios, or the fraction of time a myosin spends attached to actin during one full pass through its mechanochemical cycle. Recent work from Porter et al.[45] suggests that the conformational ensemble of the active site P-loop encodes duty ratio through the balance of nucleotide favorable and unfavorable states. Specifically, low duty motors have an increased propensity to adopt a P-loop “up” state, where the S180 carbonyl group sterically occludes nucleotide binding, whereas high duty motors favor a “down” state, where the P-loop is nucleotide compatible (see Fig. 8b).

We trained a DiffNet using molecular dynamics simulation data from the active sites of four low duty motors and four high duty motors to see if we could recapitulate the trend between P-loop dynamics and duty ratio (Fig. 8a). Importantly, this test case is especially challenging because it includes eight different proteins with a low degree of sequence conservation in the area of interest (i.e. 34% of residues were perfectly conserved within the training area). Low duty motors were given an initial label of zero and high duty motors were initially given a label of one.

A DiffNet trained to distinguish high and low duty myosin motors substantiates previous work that identified P-loop dynamics to be important for distinguishing these myosins. To determine if a DiffNet captures the importance of P-loop “up” and “down” states, we examined structures with low and high DiffNet output labels (i.e. predicted low and high duty respectively) from a single isoform. We saw a consistent trend in the orientation of the S180 carbonyl group, where structures with high DiffNets labels are in the “down” orientation and structures with low labels are in the “up” orientation (see Figure 8b). This indicates that the DiffNet correctly learned that high-duty motors are more likely to be in the “down” state and vice-versa. To more precisely quantify this trend, we examined the correlation between DiffNet output labels and nucleotide compatibility (as defined previously[45]) for all frames. We find that as the DiffNet output labels increase (i.e. shift from low duty to high duty), there is a concurrent increase in the ratio of nucleotide favorable:unfavorable states (Fig. 8c).

Automated interpretation of a DiffNet captures the importance of P-loop dynamics and suggests other order parameters that may distinguish high and low duty myosins. Similar to Figure 6, we calculate the correlation between interatomic distances and DiffNet output labels for all 139,129 distances around the active site (Figure 8a) and then project the top 100 correlated distances onto the structure (Fig. 8d). This analysis finds 78 distances between the P-loop and the loop connecting the third beta sheet with the SH2 helix (referred to as the β3-SH2 loop), again highlighting that the DiffNet learns that P-loop dynamics are important for discriminating high and low duty motors. Importantly, the DiffNet infers that high duty motors are more likely to occupy states where the P-loop is near to the β3-SH2 loop, and indeed this finding is confirmed using previously published Markov State Models of the motor domains (see Figure 8e). Since the β3-SH2 loop is below the P-loop, this provides further evidence that the DiffNet is correctly learning that high duty motors prefer the “down” state. While this order parameter is the predominant feature of this analysis, the DiffNet suggests that other distances may be important for distinguishing high and low duty motors. In particular, there are two residues on switch-II with distances that are strongly correlated with the DiffNet label indicating that conformational changes in switch-II may be important for determining the duty ratio, which is consistent with previous findings[46].

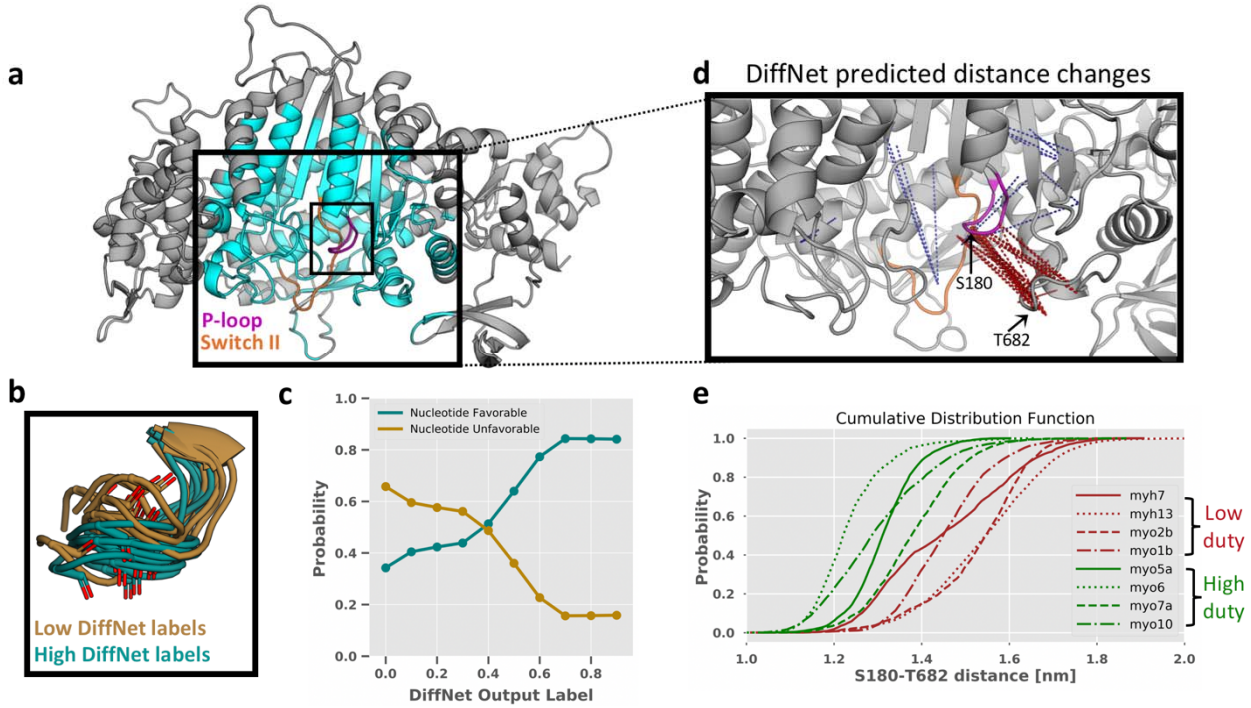


Figure 8. DiffNets capture the importance of P-loop motions in distinguishing high and low duty myosin motor proteins. **a** Structure of a myosin motor protein showing the DiffNet classification region (cyan), the P-loop (magenta), and Switch-II (Orange). **b** Twenty states predicted by the DiffNet as high duty (teal) and low duty (dark gold). Predicted high duty states are mostly in a nucleotide compatible, P-loop "down" conformation and vice-versa for predicted low duty states. **c** Percentage of nucleotide favorable (teal) and unfavorable (dark gold) states as a function of the DiffNet output label, measured with 10 equally spaced bins with labels spanning 0-1. Most structures with low Diffnets labels are nucleotide unfavorable, and vice-versa. **d** Inset of the myosin active site. Dotted lines indicate distances between two atoms that change in a way that is strongly correlated with an increased DiffNet output label. Red indicates the atoms move closer together as the output label increases, blue indicates atoms moving away from each other. **e** Cumulative distribution function showing the distance between S180 (P-loop) and T682 (β3-SH2 loop). Probabilities come from a previously published MSM[45]. This distance clearly separates high and low duty motors (green and light brown, respectively) as predicted by the DiffNet in (d).

Discussion

We have introduced DiffNets, a deep learning framework for identifying the structural signatures that are predictive of biochemical differences between protein variants from molecular dynamics simulations. Such simulations contain valuable information about the structural mechanisms that determine proteins' biochemical properties. However, extracting this insight is often difficult because of factors like the high dimensionality of the spaces involved and overlap between the structural ensembles for different variants. Our results suggest that self-supervised DiffNets learn a low-dimensional latent representation of protein structures that separates them based on their association with biochemical properties, such as higher or lower stability. This success relies on two key innovations. First, performing dimensionality reduction simultaneously with a classification task helps yield a latent representation that organizes protein structural configurations based on their association with biochemical properties. Second, challenges with

labelling each structure with a biochemical property can be overcome using an expectation maximization scheme inspired by multiple instance learning. As a proof of principle, we demonstrated that DiffNets automatically identify subtle structural differences that are predictive of the relative stabilities of different variants of the protein β-lactamase. To demonstrate the generality of the approach, we also showed DiffNets learns how duty ratio is encoded across myosin isoforms with low sequence conservation. Looking ahead, we expect the same architecture to be applicable to other perturbations, such as post-translational modifications or the presence/absence of a binding partner.

Methods

Molecular Dynamics Simulations

All molecular dynamics simulation data were generated in previous manuscripts by Zimmerman et. al[5] and Porter et. al[45]. Briefly, all simulations were run with Gromacs 5.1.1 at a temperature of 300K using the AMBER03 force field with explicit TIP3P solvent[47,48]. β-lactamase simulations were initialized from the TEM-1 β-lactamase crystallographic structure (PDB ID: 1JWP)[38] and ran at 300K using the AMBER03 force field with explicit TIP3P solvent[47,48]. Each variant, wild-type, M182V, M182T, M182S, and M182N was simulated for 6.5 μs including 4 μs of FAST-RMSD adaptive sampling[49] and 2.5 μs of conventional sampling. Conformations were stored every 20 ps. Myosin simulations were performed mostly on Folding@Home[50] to obtain ~2 milliseconds of total sampling across four low duty (MYH13, MYH7, MYH10, and MYO1B) and four high duty motors (MYO5A, MYO6, MYO7A, and MYO10), where the initial structures were built from homology models in SWISS-MODEL[51] using the 4PA0[52] as a guide template structure.

Data Processing

Simulation data was preprocessed before becoming input to the DiffNets. Simulation trajectories and the original crystallographic structure (PDB ID: 1JWP) are stripped down to the XYZ coordinates of the protein backbone without carbonyl oxygens (C, CA, CB, and N). Then, the trajectories are centered at the origin and aligned to the crystallographic structure. Next, we follow a procedure similar to Wehmeyer and Noe[27] to mean-shift the XYZ coordinates to zero, followed by whitening. First, we mean shift,

$$x^{mean-free} = \sum_{i=1}^{N_t} x_i - \bar{x}$$

where $x^{mean-free}$ is the mean-shifted trajectory of XYZ coordinates, x_i is a single frame with XYZ coordinates, $\bar{x}$ is the mean of the XYZ coordinates across all trajectories, and N_t is the number of frames in all trajectories.

Next, we whiten the data,

$$\tilde{x} = C_{00}^{-\frac{1}{2}} x^{mean-free}$$

where $\tilde{x}$ is the whitened trajectory of XYZ coordinates and C_{00} is the covariance matrix for the XYZ coordinates. Whitening decorrelates the inputs and adjusts their variance to be unity. After whitening, we use one out of every ten simulation frames for each epoch of DiffNet training. In practice, whitening and unwhitening of the data is performed on the input XYZ coordinates directly in the DiffNet with frozen (untrainable) weights. For myosin, we subsampled the data to use only one of every ten simulation frames.

Neural network training

We trained DiffNets with three loss functions to minimize protein reconstruction error (ℓ_{Recon}), minimize feature classification error (ℓ_{Class}), and minimize the correlation of latent space variables (ℓ_{Corr}).

$$\mathcal{L}_{DiffNet} = \ell_{Recon} + \ell_{Class} + \ell_{Corr}$$

The reconstruction loss term attempts to tune the network weights to properly reconstruct the original XYZ coordinates of the protein. This loss combines an absolute error (L1), which funnels reconstructions to the proper XYZ coordinates, and a mean-squared error (L2) to strongly discourage outliers. Explicitly,

$$\ell_{Recon} = \frac{1}{N_b}\sum_{i=1}^{N_b}\frac{1}{N_n}\sum_{j=1}^{N_n}\left[\left|x_{ij} - \hat{x}_{ij}\right| + \left(x_{ij} - \hat{x}_{ij}\right)^2\right]$$

where N_n is the number of output nodes (all XYZ coordinates), N_b is the number of examples in a training batch, x_{ij} is a target value (actual XYZ coordinate), and $\hat{x}_{ij}$ is the output value from the DiffNet.

The classification error is a binary cross entropy error that penalizes misclassifications by the latent space. This classification loss attempts to constrain the latent space to learn a dimensionality reduction that can also classify a biophysical feature. Explicitly,

$$\ell_{Class} = \frac{1}{N_b}\sum_{i=1}^{N_b} y_i * \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i)$$

where N_b is the number of examples in a training batch, y_i is the target value, a binary value indicating if a simulation frame has a specific feature or not, and $\hat{y}_i$ is the output of the classification layer by the DiffNet.

Finally, we include a loss function to minimize the covariance between latent space variables. This loss takes the form of

$$\ell_{Corr} = \sum_{i \neq j} Cov\left(z_i, z_j\right)^2$$

where $Cov(z_i, z_j)$ is the covariance matrix of the latent vector, Z, across all N_b samples in a training batch. We reason that preventing redundancy in latent variables should maximize the amount of information one can gain in a small number of variables. Ideally, this sets us up to use just a few latent variables and still have a rich amount of information. With fewer latent variables, models are generally more interpretable.

Our training procedure uses several training iterations to progressively build in hidden layers of the DiffNet. First, we train a minimal version of a DiffNet. Explicitly, the encoders have an input layer and a reduction layer with a four-fold reduction in variables. There is no further reduction to a bottleneck layer. Instead, the decoder takes the reduction layer as input and passes it to an output layer. Training this simplified autoencoder is an easier task than training a full DiffNet because the dimensionality reduction it performs is modest. It has ~an order of magnitude more dimensions to explain the original data compared with a true bottleneck layer. We reason that this can generate useful priors for what the reduction layer should capture. In our second pretraining procedure, we freeze those priors and add the bottleneck layer in to train the full DiffNet. Therefore, this second pretraining step concentrates its representational power on tuning how to properly reduce from the reduction layer to the bottleneck layer. Finally, we unfreeze the priors and train the full DiffNet to polish all weights. Each of these three procedures undergoes 20 training epochs. In the self-supervised setting, classification labels are updated using expectation maximization after each training epoch.

All training was performed in PyTorch 1.1[53] on an NVIDIA Tesla P100. Training on ~120,000 simulation frames of β-lactamase takes under one hour. Training with expectation maximization approximately doubled the training time for DiffNets trained on TEM. We used the Adam optimizer with a learning rate of 0.0001 and a batch size of 32. We did not perform any hyperparameter search.

Reconstruction experiment

To analyze DiffNet reconstruction error (Figure 3), we trained on five architectures where we varied the numbers of latent variables. All architectures split the input (as in Figure 1b) such that any atom (C, CA, N, CB) within 1nm of residue 182 (source of single point mutation – colored cyan in Fig 6) was included in encoder A, while the rest of the protein was included in encoder B. Encoder A reduced down to 1, 2, 3, 5, and 10 latent variables, while encoder B reduced down to 2, 3, 7, 20, and 40 latent variables. After training, we use the neural networks to reconstruct the protein structure from 1 of every 100 simulation frames and compute its root-mean squared deviation from the actual structure obtained via simulation.

Classification labels

To provide classification labels for Figure 4, we designated simulation frames as “compact helix” or “extended helix” based on a previous manuscript that identified three key hydrogen bond distances in Helix 9 that distinguish stabilizing variants from nonstabilizing variants (Res 182-186, Res 183-187, and Res 186-190)[5]. Specifically, we label helix 9 compact if the distance between the backbone nitrogen and the carbonyl oxygen is less than 4.2 Ångstroms for all residue pairs listed, and we label it extended otherwise.

β-lactamase expectation maximization experiment

When training on β-lactamase with expectation maximization (Fig 5, 6, 7) we trained a split architecture DiffNet with 30 latent variables that was split proportionally into encoder A and B based on the number of atoms input into each encoder (i.e. 4 and 26 latent variables in encoder A and B, respectively). After training, we ran DiffNets on all simulation data to obtain DiffNet output labels. We bin all structures into 10 equally spaced bins from 0-1 based on their DiffNet output label. Then, we measure the average distance for Res 182-186, Res 183-187, and Res 186-190 in each bin (Figure 5a). We calculated an AUC evaluating how well the DiffNet output labels classify compact helix 9 states from extended stated where the labelling criteria is explained in the above section. To find distance changes that are correlated with changes in the DiffNet output label (as shown in Figure 6), we first cluster the simulation data into 2000 clusters

using a hybrid k-centers and k-medoids approach with our open-source python package, Enspara[54]. Then, we enumerate all possible distance pairs between atoms in encoder A (i.e. within 1nm of the mutation). For each distance pair, we perform a linear regression between the distance and the DiffNet output label across all 2000 cluster centers. We then select the distance pairs with the highest correlation coefficients (top 1%) and visualize them in PyMol (Figure 6).

Myosin expectation maximization experiment

When training on myosin (Fig. 8) we used an architecture with a single encoder (i.e. not split) that received C, CA, N, and CB atoms as input within a 2.25 nm radius around the P-loop (specifically residue S180, *Myh7* numbering). We used 50 latent variables. All frames from low duty motors were initially assigned classification labels of 0, and simulation frames from high duty motors were initially assigned 1s. During the EM procedure, we set bounds of 10-40% for low duty motor frames and 60-90% for high duty motor frames. To find distance changes that are correlated with changes in the DiffNet output label, we copied the scheme described in the previous section. To identify P-loop orientations with high/low DiffNet labels, we selected the 10 structures with DiffNet labels closest to 0.03 and 0.7 from a single isoform (Myh7). To calculate the cumulative distribution function in Fig. 8e, we used a previously published MSM. Specifically, for each cluster center in the MSM, we measured its distance and weighted the distance by its equilibrium population. Lastly, for Fig. 8c, we grouped structures as nucleotide favorable/unfavorable as defined in a previous manuscript[45].

Code Availability. Data normalization and DiffNets training with, or without, expectation maximization is freely available on GitHub at https://github.com/bowman-lab/diffnets.

Data Availability. The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Acknowledgements

This work was funded by NSF CAREER Award MCB-1552471 and NIH grant R01 GM124007 (GRB). GRB holds a Career Award at the Scientific Interface from the Burroughs Wellcome Fund and a Packard Fellowship for Science and Engineering from The David & Lucile Packard Foundation. MDW was supported by a MolSSI COVID-19 seed software fellowship and would like to thank Sina Mostafanejad and Doaa Altarawy for their guidance in developing DiffNets as a software package.

Author Contributions

All authors conceptualized different aspects of the research and developed software. M.D.W. and G.R.B. acquired funding. M.D.W., M.I.Z., A.M., M.C. and G.R.B. conducted research and developed software. M.D.W. wrote the original draft and all other authors reviewed and edited the manuscript. M.D.W. M.I.Z., A.M., and M.C. created the data visualizations.

Competing Interests. The authors declare no competing financial interests.

References

1. Erickson RP. Somatic gene mutation and human disease other than cancer: An update. *Mutat Res - Rev Mutat Res*. 2010. doi:10.1016/j.mrrev.2010.04.002
2. Krawczak M, Ball E V., Fenton I, et al. Human Gene Mutation Database - A biomedical

information and research resource. *Hum Mutat*. 2000. doi:10.1002/(SICI)1098-1004(200001)15:1<45::AID-HUMU10>3.0.CO;2-T
3. Davies J. Origins and evolution of antibiotic resistance. *Microbiologia*. 1996. doi:10.1128/mmbr.00016-10
4. Sultan MM, Wayment-Steele HK, Pande VS. Transferable Neural Networks for Enhanced Sampling of Protein Dynamics. *J Chem Theory Comput*. 2018. doi:10.1021/acs.jctc.8b00025
5. Zimmerman MI, Hart KM, Sibbald CA, et al. Prediction of New Stabilizing Mutations Based on Mechanistic Insights from Markov State Models. *ACS Cent Sci*. 2017. doi:10.1021/acscentsci.7b00465
6. Perryman AL, Lin J-H, McCammon JA. HIV-1 protease molecular dynamics of a wild-type and of the V82F/I84V mutant: Possible contributions to drug resistance and a potential new target site for drugs. *Protein Sci*. 2004. doi:10.1110/ps.03468904
7. Schwantes CR, Shukla D, Pande VS. Markov state models and tICA reveal a nonnative folding nucleus in simulations of NuG2. *Biophys J*. 2016. doi:10.1016/j.bpj.2016.03.026
8. Sang D, Pinglay S, Wiewiora RP, et al. Ancestral reconstruction reveals mechanisms of erk regulatory evolution. *Elife*. 2019. doi:10.7554/eLife.38805
9. Razavi AM, Voelz VA. Kinetic Network Models of Tryptophan Mutations in β-Hairpins Reveal the Importance of Non-Native Interaction. *J Chem Theory Comput*. 2015. doi:10.1021/acs.jctc.5b00088
10. Romero PA, Arnold FH. Exploring protein fitness landscapes by directed evolution. *Nat Rev Mol Cell Biol*. 2009. doi:10.1038/nrm2805
11. James LC, Tawfik DS. Conformational diversity and protein evolution - A 60-year-old hypothesis revisited. *Trends Biochem Sci*. 2003. doi:10.1016/S0968-0004(03)00135-X
12. Hart KM, Ho CMW, Dutta S, Gross ML, Bowman GR. Modelling proteins' hidden conformations to predict antibiotic resistance. *Nat Commun*. 2016. doi:10.1038/ncomms12965
13. Knoverek CR, Amarasinghe GK, Bowman GR. Advanced Methods for Accessing Protein Shape-Shifting Present New Therapeutic Opportunities. *Trends Biochem Sci*. 2019. doi:10.1016/j.tibs.2018.11.007
14. Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. *Nat Struct Biol*. 2002. doi:10.1038/nsb0902-646
15. Bowman GR, Pande VS, Noé F. An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation. *Springer*. 2014. doi:10.1007/978-94-007-7606-7
16. Husic BE, McKiernan KA, Wayment-Steele HK, Sultan MM, Pande VS. A Minimum Variance Clustering Approach Produces Robust and Interpretable Coarse-Grained Models. *J Chem Theory Comput*. 2018. doi:10.1021/acs.jctc.7b01004
17. David CC, Jacobs DJ. Principal component analysis: A method for determining the essential dynamics of proteins. *Methods Mol Biol*. 2014. doi:10.1007/978-1-62703-658-0_11
18. Teodoro ML, Phillips GN, Kavraki LE. Understanding Protein Flexibility through Dimensionality Reduction. In: *Journal of Computational Biology*. ; 2003. doi:10.1089/10665270360688228
19. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science (80-)*. 2006. doi:10.1126/science.1127647
20. Naritomi Y, Fuchigami S. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions. *J Chem Phys*. 2011. doi:10.1063/1.3554380
21. Pérez-Hernández G, Paul F, Giorgino T, De Fabritiis G, Noé F. Identification of slow molecular order parameters for Markov model construction. *J Chem Phys*. 2013.

doi:10.1063/1.4811489
22. Mardt A, Pasquali L, Wu H, Noé F. VAMPnets for deep learning of molecular kinetics. *Nat Commun*. 2018. doi:10.1038/s41467-017-02388-1
23. Le L, Patterson A, White M. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In: *Advances in Neural Information Processing Systems*. ; 2018.
24. Lemke T, Peter C. EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations. *J Chem Theory Comput*. 2019. doi:10.1021/acs.jctc.8b00975
25. Hernández CX, Wayment-Steele HK, Sultan MM, Husic BE, Pande VS. Variational encoding of complex dynamics. *Phys Rev E*. 2018. doi:10.1103/PhysRevE.97.062412
26. Tsuchiya Y, Taneishi K, Yonezawa Y. Autoencoder-Based Detection of Dynamic Allostery Triggered by Ligand Binding Based on Molecular Dynamics. *J Chem Inf Model*. 2019. doi:10.1021/acs.jcim.9b00426
27. Wehmeyer C, Noé F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J Chem Phys*. 2018. doi:10.1063/1.5011399
28. Chen W, Ferguson AL. Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration. *J Comput Chem*. 2018. doi:10.1002/jcc.25520
29. Teletin M, Czibula G, Bocicor MI, Albert S, Pandini A. Deep autoencoders for additional insight into protein dynamics. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. ; 2018. doi:10.1007/978-3-030-01421-6_8
30. Wang Y, Ribeiro JML, Tiwary P. Past–future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics. *Nat Commun*. 2019. doi:10.1038/s41467-019-11405-4
31. Wang Y, Lamim Ribeiro JM, Tiwary P. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Curr Opin Struct Biol*. 2020. doi:10.1016/j.sbi.2019.12.016
32. Lusch B, Kutz JN, Brunton SL. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat Commun*. 2018. doi:10.1038/s41467-018-07210-0
33. Greener JG, Moffat L, Jones DT. Design of metalloproteins and novel protein folds using variational autoencoders. *Sci Rep*. 2018. doi:10.1038/s41598-018-34533-1
34. Degiacomi MT. Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space. *Structure*. 2019. doi:10.1016/j.str.2019.03.018
35. Noé F, De Fabritiis G, Clementi C. Machine learning for protein folding and dynamics. *Curr Opin Struct Biol*. 2020. doi:10.1016/j.sbi.2019.12.005
36. Jacquier H, Birgy A, Le Nagard H, et al. Capturing the mutational landscape of the beta-lactamase TEM-1. *Proc Natl Acad Sci U S A*. 2013. doi:10.1073/pnas.1215206110
37. Orencia MC, Yoon JS, Ness JE, Stemmer WPC, Stevens RC. Predicting the emergence of antibiotic resistance by directed evolution and structural analysis. *Nat Struct Biol*. 2001. doi:10.1038/84981
38. Wang X, Minasov G, Shoichet BK. Evolution of an antibiotic resistance enzyme constrained by stability and activity trade-offs. *J Mol Biol*. 2002. doi:10.1016/S0022-2836(02)00400-X
39. Thomas VL, McReynolds AC, Shoichet BK. Structural Bases for Stability-Function Tradeoffs in Antibiotic Resistance. *J Mol Biol*. 2010. doi:10.1016/j.jmb.2009.11.005
40. Woodford N, Ellington MJ. The emergence of antibiotic resistance by mutation. *Clin Microbiol Infect*. 2007. doi:10.1111/j.1469-0691.2006.01492.x
41. Salverda MLM, de Visser JAGM, Barlow M. Natural evolution of TEM-1 β-lactamase: Experimental reconstruction and clinical relevance. *FEMS Microbiol Rev*. 2010. doi:10.1111/j.1574-6976.2010.00222.x

42. Zaretzki JM, Browning MR, Hughes TB, Swamidass SJ. Extending P450 site-of-metabolism models with region-resolution data. *Bioinformatics*. 2015. doi:10.1093/bioinformatics/btv100
43. Moon TK. The expectation-maximization algorithm. *IEEE Signal Process Mag*. 1996. doi:10.1109/79.543975
44. De La Cruz EM, Ostap EM. Relating biochemistry and function in the myosin superfamily. *Curr Opin Cell Biol*. 2004. doi:10.1016/j.ceb.2003.11.011
45. Porter JR, Meller A, Zimmerman MI, Greenberg MJ, Bowman GR. Conformational distributions of isolated myosin motor domains encode their mechanochemical properties. *Elife*. 2020. doi:10.7554/eLife.55132
46. Llinas P, Isabet T, Song L, et al. How Actin Initiates the Motor Activity of Myosin. *Dev Cell*. 2015. doi:10.1016/j.devcel.2015.03.025
47. Duan Y, Wu C, Chowdhury S, et al. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-Phase Quantum Mechanical Calculations. *J Comput Chem*. 2003. doi:10.1002/jcc.10349
48. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML. Comparison of simple potential functions for simulating liquid water. *J Chem Phys*. 1983. doi:10.1063/1.445869
49. Zimmerman MI, Bowman GR. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *J Chem Theory Comput*. 2015. doi:10.1021/acs.jctc.5b00737
50. Shirts M, Pande VS. Screen savers of the world unite. *Science (80-)*. 2000. doi:10.1126/science.290.5498.1903
51. Waterhouse A, Bertoni M, Bienert S, et al. SWISS-MODEL: Homology modelling of protein structures and complexes. *Nucleic Acids Res*. 2018. doi:10.1093/nar/gky427
52. Winkelmann DA, Forgacs E, Miller MT, Stock AM. Structural basis for drug-induced allosteric changes to human β-cardiac myosin motor activity. *Nat Commun*. 2015. doi:10.1038/ncomms8974
53. Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. In: *Advances in Neural Information Processing Systems 32*. ; 2019.
54. Porter JR, Zimmerman MI, Bowman GR. Enspara: Modeling molecular ensembles with scalable data structures and parallel computing. *J Chem Phys*. 2019. doi:10.1063/1.5063794