JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

Construction of a Deep Neural Network Energy Function for Protein Physics

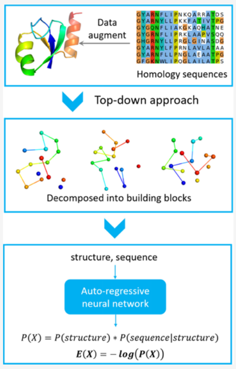
Huan Yang,* Zhaoping Xiong, and Francesco Zonta*

Cite This: *J. Chem. Theory Comput.* 2022, 18, 5649−5658

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The traditional approach of computational biology consists of calculating molecule properties by using approximate classical potentials. Interactions between atoms are described by an energy function derived from physical principles or fitted to experimental data. Their functional form is usually limited to pairwise interactions between atoms and does not consider complex multibody effects. More recently, neural networks have emerged as an alternative way of describing the interactions between biomolecules. In this approach, the energy function does not have an explicit functional form and is learned bottom-up from simulations at the atomistic or quantum level. In this study, we attempt a top-down approach and use deep learning methods to obtain an energy function by exploiting the large amount of experimental data acquired with years in the field of structural biology. The energy function is represented by a probability density model learned from a large repertoire of building blocks representing local clusters of amino acids paired with their sequence signature. We demonstrated the feasibility of this approach by generating a neural network energy function and testing its validity on several applications such as discriminating decoys, assessing qualities of structural models, sampling structural conformations, and designing new protein sequences. We foresee that, in the future, our methodology could exploit the continuously increasing availability of experimental data and simulations and provide a new method for the parametrization of protein energy functions.


Data augment
Homology sequences
Top-down approach
Decomposed into building blocks
structure, sequence
Auto-regressive neural network
$P(X) = P(structure) * P(sequence|structure)$
$E(X) = -log(P(X))$

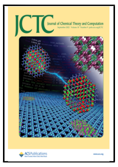
■ INTRODUCTION

A realistic description of biological molecules should involve Quantum Mechanics (QM); however, its computational cost strongly limits feasible applications of such an approach. Models based on approximate classical potential, on the other hand, have been more successful in describing how proteins and other biological molecules interact. Two main categories of simplified potentials have emerged: physics-based potentials and knowledge-based potentials. Both methods rely on physical intuition to map interactions between atoms into simple functional forms that depend on a set of parameters. In the first case, the parameters are derived by comparison with high-precision QM calculations or experiments. These potentials often take the name of force fields and are mainly used to perform molecular dynamics (MD) simulations.[1−5] In the second case, probability distributions of observables (e.g., distances, angles, native contacts, etc.) are directly obtained from experimental information and transformed into a statistical potential.[6−10] These potentials are mainly used for applications that are too computationally expensive for MD simulations. The two approaches are not antithetic, and many hybrid potentials have been developed from combining the two methodologies.[11,12]

Despite their wide usage and success, approximate classical potentials still present major limitations that go in opposite directions. For some applications, their simple functional form cannot take into account all the details required for an accurate description of the system. At the same time, simulations based on classical potentials can be too computationally expensive and cannot always reach time and size scales directly comparable with experiments. Although QM calculations and hybrid methods[13−16] can be used to improve the accuracy of the calculations, and enhanced sampling methods[17] or coarse-grained force fields[18−21] can be used to scale up in both time and size scales, the core problems still exist.

Deep learning has entered the field of structural biology with a number of different applications that steadily increased with the years.[22,23] In most cases, deep learning has been used to complement standard bioinformatics techniques based on sequence analysis, for example, protein structure prediction using primary sequences. This classic problem, the holy grail of computational biology, has de facto been solved recently after the impressive results obtained by Alphafold2, Rosettafold, and others in the 14th Critical Assessment of Techniques for Protein Structure Prediction (CASP14) competition.[24−27] Researchers have also applied unsupervised deep learning methods to extract biological, biophysical, and evolutionary information from protein sequences.[28−30] Despite these

Received: January 20, 2022
Published: August 8, 2022



https://doi.org/10.1021/acs.jctc.2c00069

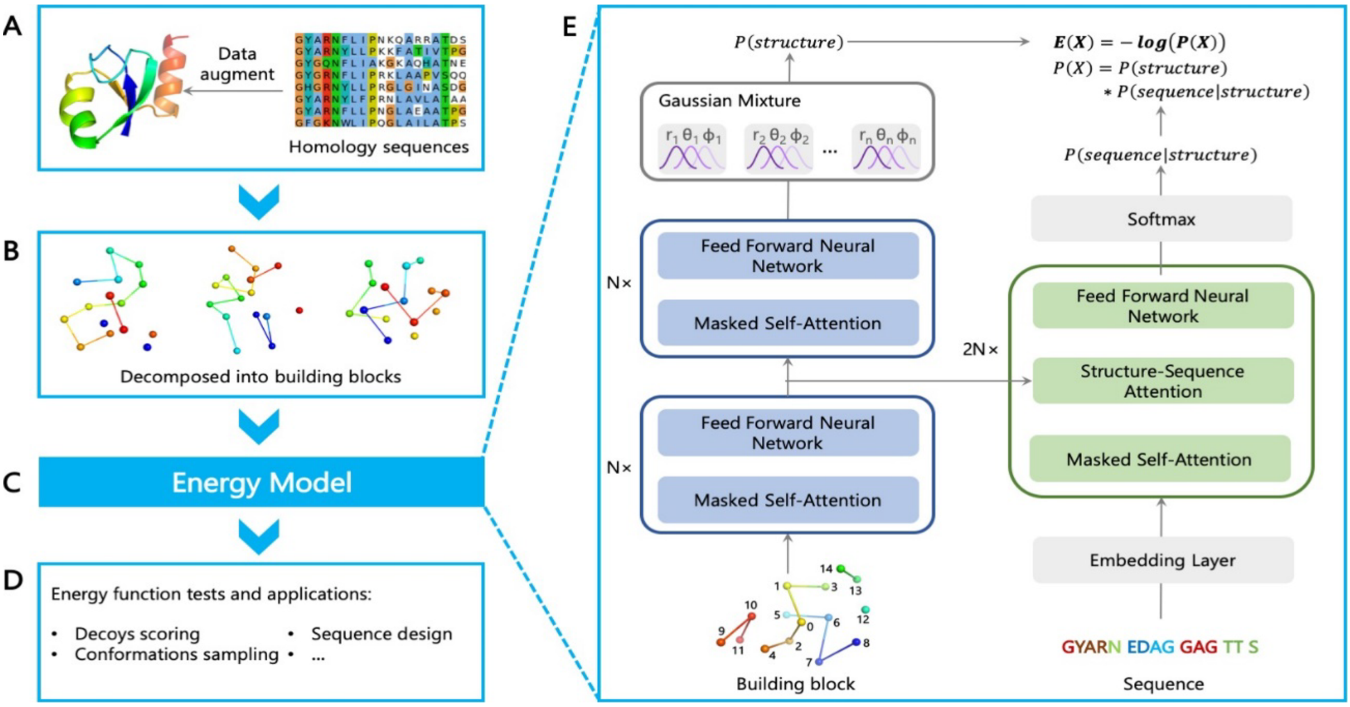


Figure 1. Overview of the NNEF model. (A) We use a sample of nonredundant protein structures and augment the training data with their homologous sequences. (B) Local structure around each residue defines the building block and is used as the input of the energy model. Each building block X includes the residue itself, its four nearest residues along the sequence, and the ten other nearest residues in the 3D space. Each residue is represented as one bead. The number of building block used as samples to train the network is about 2.5 billion. (C) The energy model is illustrated in (E), where we fit a probability density function and calculate the energy as $E(X) = -\log P(X)$. The total energy of the protein is the sum of energies of all building blocks. In the auto-regressive model, we separate the structural and sequence features and calculate $P(X) = p(\text{Structure}(X)) \cdot p(\text{Sequence}(X) \mid \text{Structure}(X))$ using the transformer network architecture. We use the *softmax* function to calculate the probabilities of discrete features and use Gaussian mixtures to calculate the probabilities of continuous features. The total number of parameters of the network model is 2.1 million. (D) The learned energy function can be applied for various tasks, such as decoy scoring, conformation sampling, and sequence design, without being explicitly trained for any of those.

successful examples, deep learning models are very data-hungry and, as a consequence, their field of application is still limited.

In more recent years, novel approaches that combine deep learning with physics-based methodologies emerged. For example, in chemistry and material science, neural network and quantum Monte Carlo methods have been applied to solve the Schrödinger equation[31,32] or generate classical atomistic force fields by fitting calculations based on density functional theory[33,34] or ab initio molecular dynamics.[35,36] In protein science, some groups have developed neural network energy functions for multiscale modeling of proteins, following a bottom-up approach, for example, fitting coarse-grained potentials to atomic MD simulations.[37] Even though these energy functions are not transferable to different protein systems, they could lead to a paradigm shift in how force fields will be parametrized in the future.

In this study, we propose a top-down approach and use unsupervised deep learning to construct a statistical potential from experimental data (protein structure and sequences). The statistical potential has not a simple functional form, but it is instead represented by a deep neural network. We then treat this statistical potential as a proper energy function (neural network energy function or NNEF) that can be applied to various tasks in pair with other standard methodology (for example, running MD simulations or discriminating decoys from native configurations).

In strict terms, the NNEF is a free energy function that takes into account both enthalpic and entropic contributions. Because entropy is not explicitly considered, hereafter, we use the term “energy function” for simplicity.

The energy function should depend only on the protein configuration, be time-independent and differentiable, and be invariant under rotations, translations, and permutations of the components to reproduce protein physics correctly.

■ RESULTS

Design of the Neural Network Energy Function Based on Protein Building Blocks. In constructing the data-driven energy function for proteins, the choice of the training data sets is critical, and a naïve training of the network with all the known experimental protein structures and sequences will likely fail to reproduce protein physics correctly. In an ideal case, we would like to have a large sample of unbiased sequence-structure pairs. In reality, when one looks at their physical properties, the set of proteins with known experimental structures is small and redundant. The ensemble of known folding domains[38,39] is likely incomplete as structures that appear legitimate from an energetic point of view may have not yet been selected or explored by evolution.[40] However, we can hypothesize that evolution had enough time to extensively explore the configuration space of smaller three-dimensional (3D) local structures.[41−43] Such local structural patterns have been chosen not only for their biological meaning but also because they have lower energy than a random configuration of the same group of amino acids. Under this hypothesis, all the information necessary to build a working energy function is already contained in the available protein databases.

Following these guidelines, we decided to represent a protein as a collection of small-scale structures (building blocks) paired with their sequence information. Each building block describe the chemical environment around a single amino acid. Building blocks can overlap as a certain amino acid can belong to different building blocks. In the training phase, the building blocks are selected from a nonredundant set of protein structures and their homologous sequences (see Figure 1 and Methods). The neural network learns the probability $P(X)$ of a given building block X from its structural and sequence features as $P(X) = p(\text{Structure}(X)) \cdot p(\text{Sequence}(X) \mid \text{Structure}(X))$, and we calculate the energy of the building block as the opposite of the logarithm of the probability: $E(X) = -\log P(X)$. We assume that the set is complete, that is, building blocks absent in this set are improbable and thus have higher energies.

Each building block is formally independent from the others, so that, by construction, the probability of a given protein configuration (Y) is the product of the probabilities of the single building blocks (X_i): $P(Y) = \prod_{X_i \in Y} P(X_i)$, and its energy is the sum of the energies of each building block: $E(Y) = \sum_{X_i \in Y} E(X_i)$. Interactions between different amino acids are kept in a self-consistent way, as changing the coordinates or the sequence of a single amino acid will affect all the building blocks in which the amino acid is present.

The energy function obtained in this way will be local and additive by construction. If properly trained, the neural network will be able to recognize alternative local minima, which corresponds to alternative configurations, active or intermediate states.

Protein Building Blocks. Definition and representation of building blocks rely on human intuitions. They can be pairs of residues, peptide fragments, groups of residues, and so forth. They can be represented by atoms or virtual beads paired with various geometric and chemical features. Furthermore, the size of the building blocks has to be small enough to minimize the risk of learning from undersampled sets but large enough to be able to represent complex tertiary structures.

In our current model, we define a building block as the local structure of 15 amino acids, which includes a central residue, its four nearest neighboring along the sequence, and the ten other residues which are closest to the central one in the 3D space (Figures 1B and S1). With this definition, building blocks are typically composed of a few noncontiguous segments of residues that can be far in the primary sequence. As we will show, the energy function can be used to run MD simulations, and in this particular case, the building blocks change dynamically with time. To represent the structure of building blocks, we use very simple low-resolution geometric features, that is, the coordinates of beads at the C_β positions (C_α for Glycine) and the connectivity of these beads based on the protein sequence. Other features, such as the positions of backbone atoms, the side chain positions, partial charges, hydrogen bonds, and so forth, could be added in future refined versions of the energy function but have not been considered in the current realization. The coordinates of all the beads composing each building block are rotated to the same local internal coordinate system (Figure S2) to guarantee rotational–translational invariance of the energy function.

Understanding the Neural Network Energy Function. To understand whether the NNEF can correctly grasp the physical properties of proteins, we test it in several tasks described hereafter.

Scoring Decoys Generated by Modifications of the Native Structures. A good energy function should be able to discriminate native-like from non-native conformations. We tested the energy function against the 3DRobot decoy set[44] and a second set of decoys that we generated by sampling normal modes of protein structures. The 3DRobot decoy set includes 200 nonhomologous single-domain proteins (48 in the *alpha* class, 40 in the *beta* class, and 112 in the *alpha/beta* class) and 300 structural decoys for each of these proteins, with root-mean-square deviation (RMSD) ranging from 0 to 12 Å. The second set was generated from 18 small (<120 residues) proteins, comprising 4 *alpha* proteins, 7 *beta* proteins, and 7 *alpha/beta* proteins. We can observe that for all the proteins we considered, the energy values tend to increase with the distance from the native, with native-like decoys having low energies and decoys with a large RMSD having higher energies (a typical result for each set is shown in Figure 2).

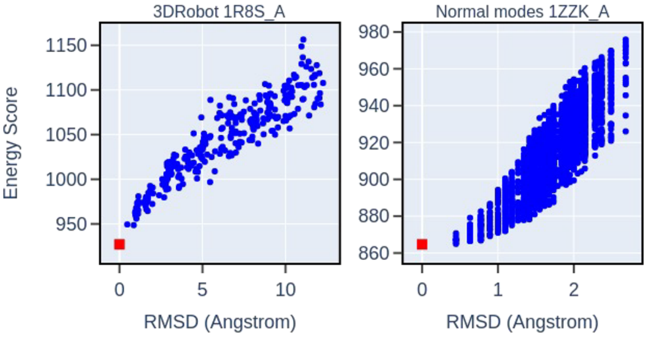


Figure 2. Scoring decoys generated by modifications of the native structures. Left panel shows one typical example protein in the 3DRobot decoy set. Right panel shows one typical example protein in the normal mode decoy set. Red square is the native structure, and blue dots are decoys. In all proteins of both decoy sets, the energy increases with the distance from the native.

Scoring Structure Predictions. Another way in which we tested the quality of the NNEF is to score predictions generated in the CASP14 contest. This is a more significant challenge than scoring the decoy sets generated by modifications of native structures because such predicted structures cover more diverse conformations and are optimized according to some other scoring functions. Each of the 97 protein domains in CASP14 has about 200–500 model predictions submitted by different groups. We evaluated the NNEF for each of these predictions and measured its correlation with the CASP14 Global Distance Test Total Score (GDT_{TS}, a measure of the overall quality of the prediction). For about 70% of the sets, we obtain a Pearson correlation coefficient $|\rho| > 0.75$ (Figure 3A,B). In the remaining cases (many of which are proteins that belong to complexes and for which their tertiary native structures could depend on the environment), the energy of the best CASP prediction is always near the global minimum of the evaluated NNEF for the whole set. In other words, the energy function appears to be quite successful in detecting good configurations but could be fooled by wrong but reasonable configurations (Figure 3C,D).

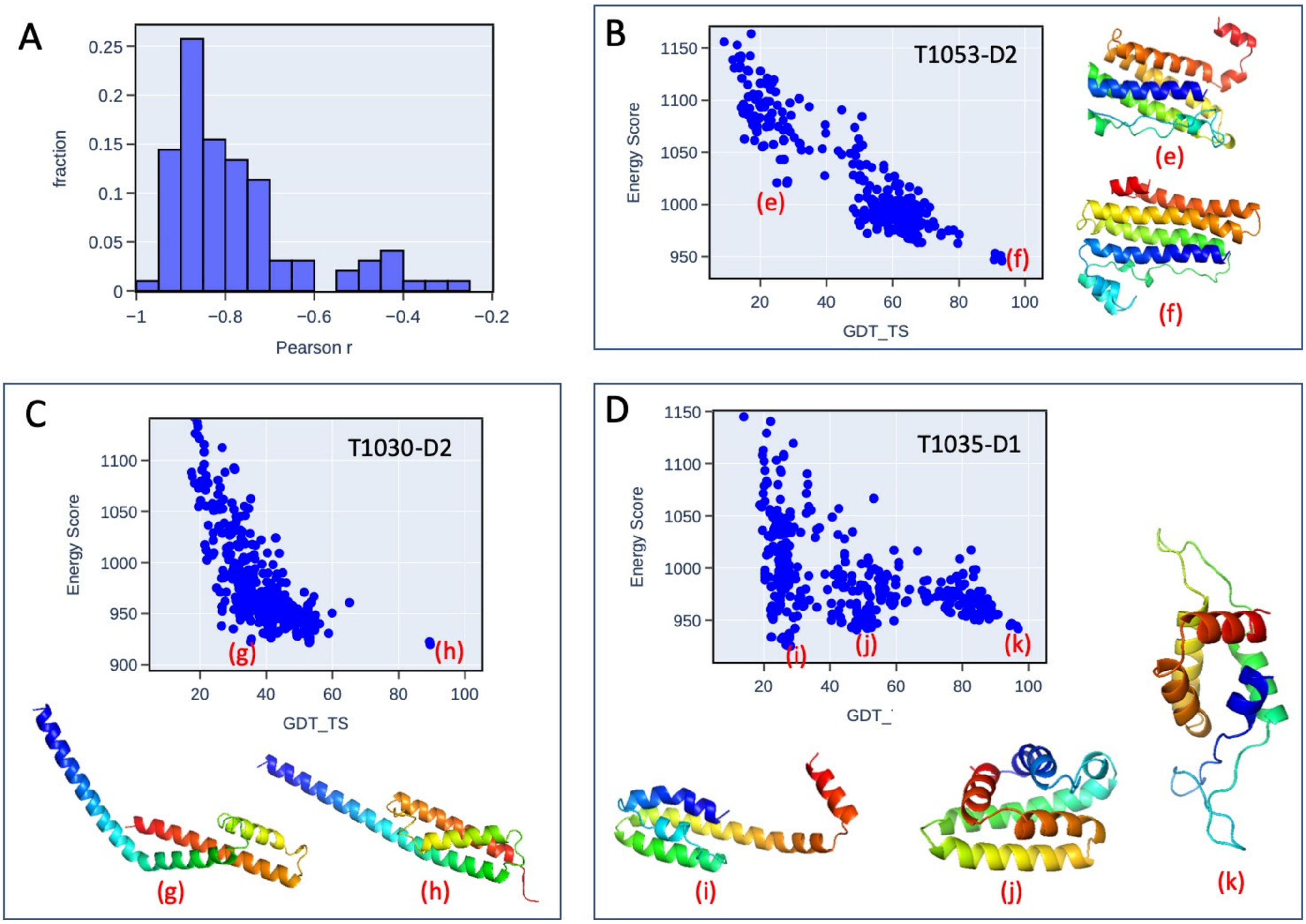


Figure 3. Scoring structure predictions in CASP14. Panel (A) shows the histogram of Pearson correlation coefficients ρ between the energy score and CASP GDT_TS score for proteins in CASP14. About 70% of proteins have Pearson correlation coefficients $|\rho\ | > 0.75$. Panels B, C, and D show three particular cases. 3D structures of some decoys are shown with indicators of their positions in the plot of energy vs GDT_TS. Panel (B) shows an example of good correlation between the NNEF energy and CASP GDT_TS score. Panel (C) shows an example of a protein with a simple *alpha*-helices fold. In this case, some models with non-native helix organizations have comparable energies to native-like models. Panel (D) shows an example of a protein involved in a complex context. Some models with wrong folds have energies lower than those of the native-like models.

Evaluating the Energy of Configurations within a MD Trajectory. The results above suggest that the learned energy function can be generalized to non-native configurations, despite being trained only with native structures. To further explore this feature, we use the NNEF to score a conformation ensemble of a small protein (Fip35) sampled over a 100-microsecond long MD trajectory.[3] The simulation was performed at 395 K using the Amber99SB force field,[45] and the trajectory is publicly available for download. The temperature of the simulation approximates the protein's in silico melting temperature; therefore, Fip35 explores regions of the phase space far from the native configuration and undergoes multiple folding and unfolding events within the simulated time window. We computed the RMSD along the original MD trajectory and compared it with the energy evaluated with the NNEF. As shown in Figure 4, our results show that the folded states have low energy scores, while the unfold states generally have higher energy scores. This indicates that the NNEF is able to distinguish real configurations of proteins out of equilibrium from configurations that are more native-like.

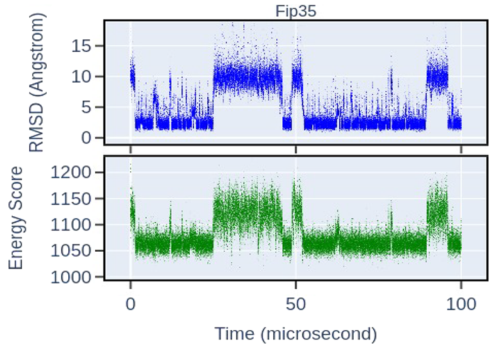


Figure 4. Evaluating the energy of configurations within a MD trajectory for a small protein Fip35. The protein undergoes multiple folding and unfolding events. The RMSD and the energy along the MD trajectory correlate well, suggesting that the energy function can generalize to non-native conformations.

Performing MD Simulations. The energy function can be interpreted as a Hamiltonian function and can be used to study protein dynamics with an implicit solvent (Langevin dynamics). In the dynamics, the Cartesian coordinates ($\vec{q}_i$)

of the residue beads are updated at each step: $\vec{q}_i(t+1) = \vec{q}_i(t) - \alpha \cdot \nabla E_i + \beta\Gamma(t)$, where $\vec{q}_i$ is the position vector of the i-th residue, E_i is the NNEF for the protein at time t, and $\Gamma(t)$ is Gaussian noise with null average and unitary variance. The coefficients α and β are related to the physical friction coefficients, the integration time step, the temperature of the system, and the physical units of energy values. With a proper choice of their values (see Methods), we can observe that the dynamics generated using this method produce fluctuations consistent with those obtained with classical MD based on force fields. We generated a 30,000 steps Langevin dynamics for each protein in a test set comprising 18 small proteins (using for all simulations the same pair of values α and β) and compared such trajectories to simulations obtained with *amber14SB* force field[5] using *OpenMM*.[46] We can observe that for most proteins in the sample, the two dynamics produce RMSFs in good correlation across the whole sequence (see Figures 5 and S4), and their difference remains below 0.5 Å. A complete assessment of the validity of the NNEF as a force field requires and deserves a much more exhaustive analysis, which is out of the scope of this article. However, these preliminary results indicate that the NNEF correctly describe a protein behavior when it is excited by thermal fluctuation.

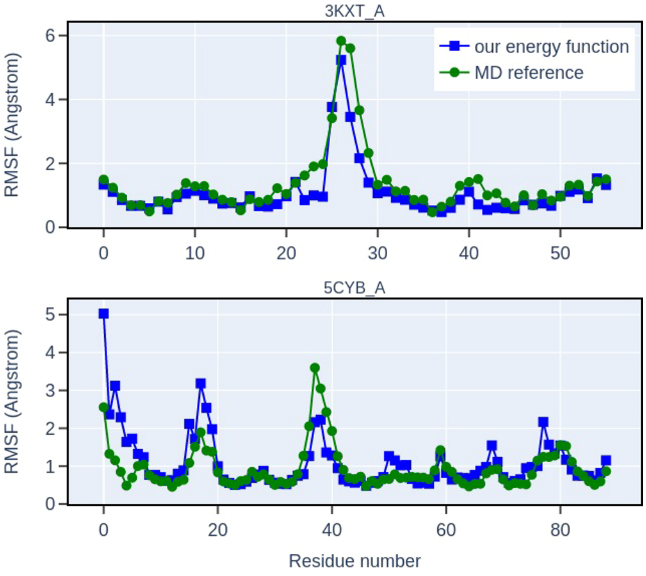


Figure 5. RMSFs resulted from Langevin dynamics simulations using the NNEF. We compare the RMSFs of the trajectories from the NNEF (blue squares) with the RMSFs of the trajectories from classical MD simulations obtained with *amber14SB* force field (green dots). Two examples with high correlation are shown here. For most proteins in the sample, the two dynamics produce RMSFs in good correlation across the whole sequence (see also Figure S5).

Importance of the Sequence in the Energy Function. The next question we want to address is whether the neural network has learned the chemical differences between amino acids and not only to evaluate local structural patterns common to any generic protein chain. This is not a given because it is possible to construct protein models that correctly describe secondary structures without any sequence information.[47] To investigate this point, we evaluated the energy of various ″decoy″ sequences assigned to the same 3D structure (100 different protein structures were used as a test). The decoy sequences were generated in four different ways: (A) substituting residues with chemically similar residues, (B) shuffling the residues in the sequence, (C) mutating residues to random ones, and (D) mutating all residues to the same residue type. In most cases, the mutated sequences' energies are higher than the correct one, and random mutations are worse than mutations to amino acids with similar chemical properties (Figure 6). However, the network appears to prefer sequences rich in alanine as almost all poly-alanine decoys have better energy than the natural sequence (Figure S3). To a lesser extent, this happens to poly-leucine and poly-histidine decoys, but not for the other amino acid.

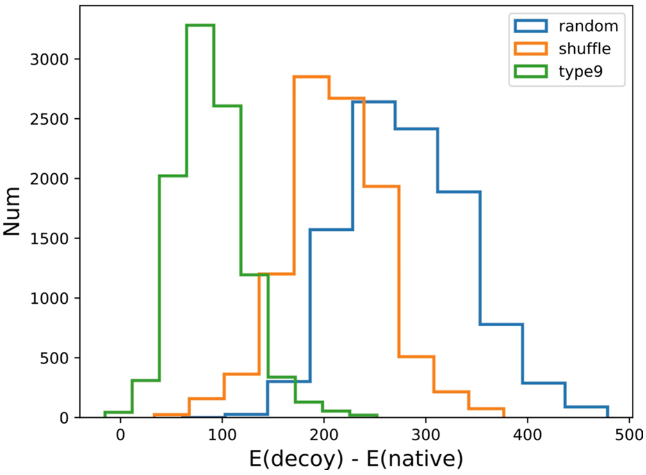


Figure 6. Distributions of energy differences E(decoy)-E(native) for three sequence decoy dataset. The decoy sequences were generated in three different ways: (1) substituting residues with chemically similar residues (green histogram), (2) shuffling the residues in the sequence (yellow histogram), and (3) mutating residues to random ones (blue histogram). The energies of the mutated sequences are higher than those of the native ones, and random mutations produce higher energies than mutations to similar amino acids.

Protein Sequence Design. To further explore the interplay between the sequence and structure in the NNEF, we redesigned the sequences of the 18 proteins in the test sample starting from the 3D conformation of their backbone. Given a random sequence, we run simulated annealing in the sequence space and attempt to minimize the energy while keeping fixed the reference structure. In this way, we expect to obtain a sequence that will eventually fold to the desired 3D configuration. At each step of the annealing process, we propose a random point mutation for the protein sequence. The mutation is accepted or rejected according to a Metropolis algorithm. In most cases, the simulations converge to sequences having energies lower than the native after a few thousand steps. We designed 100 sequences for each target protein within the mentioned test sample. Amino acid frequencies for these 1800 designed sequences are shown in Figure 7A. As we can observe, the annealing process converges to sequences that favor ten amino acids (Ala, Val, Leu, Gly, Pro, Ser, Thr, Arg, Glu, and Asp). It is worth noticing that these amino acids have relatively high frequencies in natural protein sequences and could be the first that joined biological proteins early in evolution, according to the theory on the temporal order of amino acids in evolution.[48] Overall, the average sequence recovery fraction is about 25% for the total sample of 1800 designed sequences. We also analyzed the

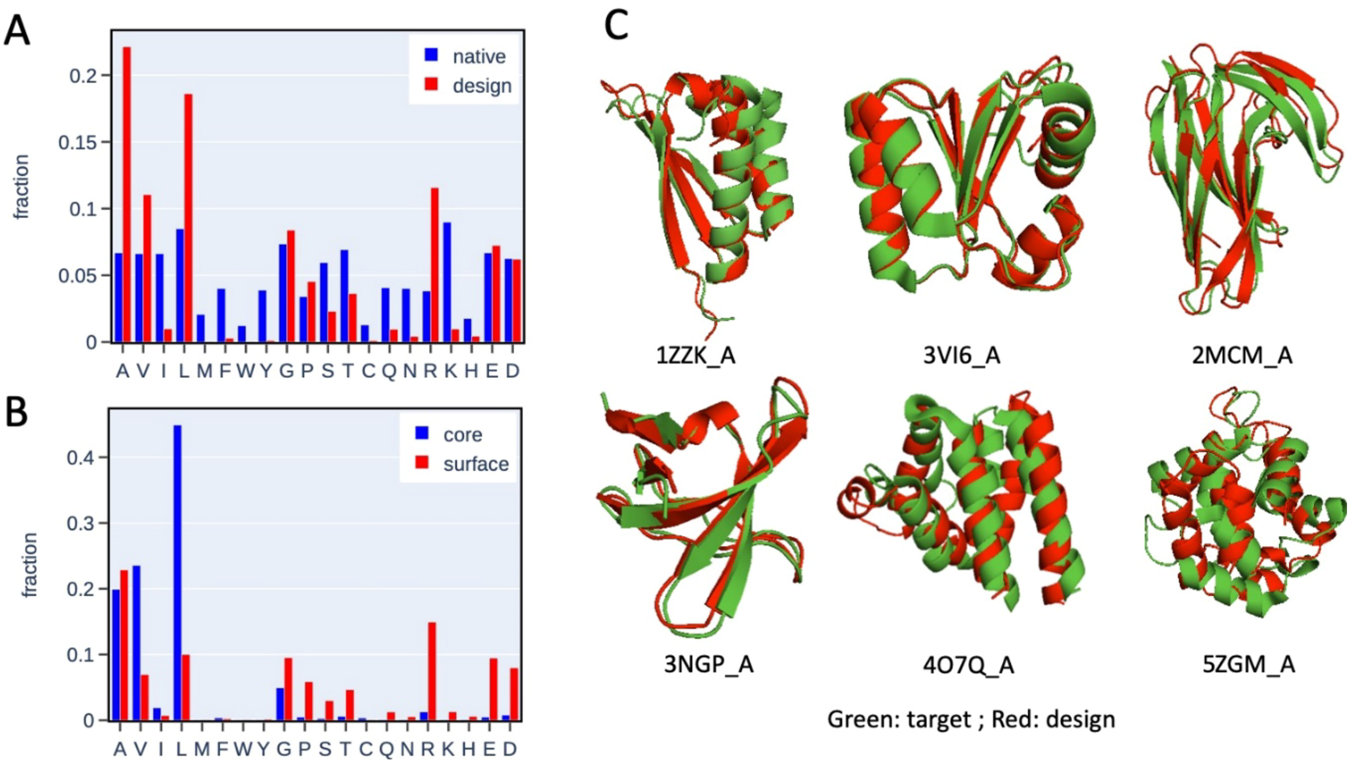


Figure 7. Results of the protein sequence design given the backbone structure. (A) Amino acid frequencies of the 1800 designed sequences and the native sequences for the test sample of 18 proteins (100 sequences for each protein). Designed sequences show a preference for ten amino acids (Ala, Val, Leu, Gly, Pro, Ser, Thr, Arg, Glu, and Asp). (B) Amino acid frequencies of the residues in the core and surface of the designed proteins. Core residues are mostly hydrophobic (Ala, Val, and Leu), while the polar residues are mainly on the surface. (C) A few examples of the target structures and predicted structures using *TrRosetta*. For about two-thirds of the designed sequences, the predicted structures can match the target structures.

differences between the core and surface residues of the designed sequences (Figure 7B). Core residues are mostly hydrophobic (Ala, Val, and Leu), while the polar residues are more likely to be exposed to the solvent. Moreover, an analysis of the 100 sequences designed for each structure reveals that core residues are more conserved than surface residues. Finally, to measure the reliability of the protein design method, we randomly selected two of the low energy designed sequences for each protein and predicted their structures using *TrRosetta*.[25] As Figures 7C and S5 show, the predicted structures match the target structures well in two-third of the cases.

DISCUSSIONS AND CONCLUSIONS

In this study, we used unsupervised deep learning methods to derive a statistical potential that describes amino acid interactions within proteins. We represented the protein as a collection of building blocks comprising nearby residues in the 3D space. The underlining ansatz is that evolution extensively explored the configuration space of such building blocks, and, for this reason, known experimental structures contain a complete and nonredundant set of building blocks that can be used to train the neural network correctly. In this way, we can get a potential that well approximates protein physics, starting from a small number of PDB structures, even though the network is defined by a large set of parameters. The strength of our methodology largely depends on the validity of our starting hypothesis. The fact that we can obtain reasonable results confirms our intuition a posteriori.

By substituting the energy function with a neural network, we are not explicitly fixing its functional form. Therefore, we can keep track of nonobvious correlations and complex multibody interactions and hopefully overcome some of the limitations of classical potentials, which are usually defined by a few pairwise interaction parameters. Indeed, the protein is not represented only by the Cartesian position of its component (in our cases the C_β atoms), but more exactly by a series of vectors that encode all the information on how that particular atom (in our case, the whole amino acid) behaves in the environment created by the other components.

For this reason, we are not restrained to a single resolution in the choice of the degree of freedoms. The protein description could be fully atomistic, coarse-grained, or mixed without any loss of generality. The network will likely be able to learn how different resolutions should merge. It is foreseeable, in the future, to use a neural network energy function to build models in which some parts have the desired level of details while other, less interesting, parts are represented at a lower resolution.

Most importantly, the NNEF is general, and it can discriminate decoys, assess qualities of structural models, sample structural conformations, and design new protein sequences, without being trained for any of these tasks specifically, indicating that it is possible to learn general properties of protein physical and chemical properties from structural data alone.

The methodology we adopted is very flexible, and our current implementation is far from realizing its full potential. We can improve the method by extensively testing different ways to combine structures and sequences, testing other representations of the proteins and building blocks and improving training methods and loss functions. It is also possible to refine the energy function parameters after the unsupervised training by applying supervised fitting to the various tasks we are interested in. Furthermore, the constantly increasing availability of experimental data could be sufficient to improve the new version of the NNEF in the future.

■ METHODS

Structure and Sequence Dataset. In order to successfully train the network, we want to curate a nonredundant sample of structures and associate each structure to a family of homologous sequences. To reduce the redundancy in the structures, we use a sample of *PDB* chains in a version of the *PISCES CulledPDB*,[49] in which the percentage identity cutoff is 50%, the resolution cutoff is 3.0 Å, the R-factor cutoff is 1.0, and the total number of chains is about 29,000. In our work, only structures solved by X-ray crystallography are included; however, it is possible to augment the structural part of the data by including also high-accuracy structure predictions.[27] Then, all the chains are matched to the *HH-suite PDB70* database[50] to get the aligned sequence data. The number of matched chains is about 19,000. We filter the aligned sequences using *hhfilter*, requiring <50% sequence identity to the *PDB* sequence and >50% sequence coverage. After filtering, we generate homologous structures by simply mapping the aligned sequences to the coordinates of the structure. Given a sequence A in a *PDB* chain and an aligned homologous sequence B, we ignore insertions in sequence B and substitute gaps in sequence B with the aligned sites in sequence A. In this way, the resulting chimeric sequence has the same length as sequence A and can be mapped to the coordinates of A.

To test the transferability of the energy function, we use a radical partition of the dataset. All chains are matched to the structural classifications in the CATH 4.2 database.[38] Each chain can include more than one CATH domain. A chain is classified as one class (e.g., *alpha/beta*) if all the CATH domains in the chain are classified as that class (*alpha/beta*). The training data include only the *alpha/beta* chains. We obtain about 7500 *alpha/beta* chains and use 7000 chains as the training dataset and 500 as the validation dataset. The trained energy function is tested on all protein classes, including *alpha/beta*, mainly *alpha* and *beta* proteins.

Neural Network Model. After decomposing the proteins of the training set into building blocks, we characterize the likelihood of a building block by fitting a probability density function and setting for each building block X: $E(X) = -\log P(X)$. The total energy of a protein Y is the sum of all building blocks composing Y: $E(Y) = \sum_{X_i \in Y} E(X_i)$. The probability function $P(X)$ is obtained with an autoregressive model and maximum-likelihood training from a set of N building blocks $\{X\}$. Each building block X can be viewed as a list of k variables $x_1, x_2, ..., x_k$ ordered in a given scheme, and its probability $P(X)$ can be expanded according to the Bayesian rule as $P(X) = \prod_j p(x_j \mid x_{<j})$. In the ordering scheme, we separate the structural and sequence features so that $P(X) = p(\text{Structure}(X)) \cdot p(\text{Sequence}(X) \mid \text{Structure}(X))$. With $P(X)$ expanded as a chain of conditional probability functions, each conditional probability $p(x_j \mid x_{<j})$ is represented as a neural network that shares parameters with other conditional probability functions. Because proteins can be represented as molecular graphs, we use transformer graph neural networks as the autoregressive model. The neural network architecture of the autoregressive model is shown in Figure 1E.

Ordering Scheme in the Autoregressive Model. In the autoregressive model, the structural and sequence feature variables are ordered according to the following rule, schematically shown in Figure S1. Each building block X can be viewed as a graph with the central bead a as the root node. The central segment is visited first with the order (a, a-1, a+1, a-2, a+2). Then, the other segments are visited in the order of increasing distance to the central bead. Within each surrounding segment, the beads are visited in the order of increasing primary sequence numbers.

Input Features. In our model, the features of a building block include the residue types, Cartesian coordinates, and bond connections of 15 amino acids belonging to a building block. To make the energies rotational−translational invariant, the coordinates of the C_β atoms are rotated to the same local internal coordinate system of the central residue (Figure S2). The coordinate system is defined so that the residue a-1 is on the X-axis, and the residues a-1 and a+1 are on the X-Y plane. The Cartesian coordinates of all beads are then converted to Polar coordinates. The bond connections, residue types, and positional labels 1−15 are converted to high dimensional vectors through look-up tables. The components of each vector are learned by the neural network and encode information about sequence and structural features of the building block. After this step, the coordinates, bond connections, and positional labels are concatenated as structural features, while residue types and positional labels are concatenated as sequence features.

Transformer Encoder and Decoder. The encoder for structural features has four standard transformer encoder layers. The outputs after two encoder layers are used as the latent codes of the structural features and passed to the decoder. The decoder for sequence features has four standard transformer decoder layers. Both the encoder and decoder layers use the standard transformer layer.[51] In the decoder, structure−sequence attention is used because we decompose the probability as $P(X) = p(\text{Structure}(X)) \cdot p(\text{Sequence}(X) \mid \text{Structure}(X))$. The attention in both the encoder and decoder is masked by position-based causal masks, that is, each position can only pay attention to positions before it, so that the network cannot know the answers by looking at the whole input data.

Neural Network Training. We use the *softmax* function to get the probabilities for the predictions of the discrete labels, such as the bond connections and the residue types. For the predictions of the continuous variables, such as the radius and angles in the coordinates, we calculate the probabilities using sums of Gaussian functions: $p(x) = \sum_j c_j G(x, \mu_j, \sigma_j)$, where $G(x, \mu_j, \sigma_j)$ is a Gaussian function of the variable x with average μ_j and variance σ_j, and c_j, μ_j, σ_j are outputs of the neural network. Thus, the network calculates the conditional probabilities of the next residue coordinate, given the coordinates of previous residues. It also calculates the conditional probabilities of the next residue type, given previous residue types and the coordinates of all residues. After getting the conditional probabilities, we calculate the energy terms as the negative log of the probabilities. The energy of the building block is the weighted sum of all the energy terms. We train the network to minimize the energies, that is, the loss function is simply the energy values. We use minibatch training, Adam optimizer[52] with starting learning rate 5×10^{-5} and *betas* = (0.9,0.99), and L2 regularization with weight 10^{-6}.

Langevin Dynamics. In the Results section, we will show that we can use the NNEF to run Langevin dynamics according to the following equation: $\vec{q}_i(t+1) = \vec{q}_i(t) - \alpha \cdot \nabla E_i + \beta\Gamma(t)$, where q_i→ is the position vector of the i-th residue, E_i is the NNEF for the i-th residue at time t, $\Gamma(t)$ is a Gaussian noise with null average and unitary variance, and α and β are physical parameters of the simulation. To decide proper values for these two parameters, we run a grid search using short simulations (α = [1e-3, 3e-3, 5e-3, 7e-3, 0.01, 0.015, 0.02, 0.04 = [0.01, 0.03, 0.06, 0.1, 0.15]). When α and β are small, the dynamics are very slow, and the protein's residues are almost locked in the initial position. When β is large, the protein unfolds after a small number of time steps quickly. We decide to use fixed values in the middle (α = 0.01 and β = 0.05) to run simulations.

Small Protein Test Set. For testing sequence reconstruction and Langevin dynamics, we used a test set of 18 small proteins corresponding to the following PDB IDs (the underscore identifies the protein chain): 1ZZK_A, 2MCM_A, 2VIM_A, 3FBL_A, 3IPZ_A, 3KXT_A, 3NGP_A, 3P0C_A, 3SNY_A, 3SOL_A, 3VI6_A, 4M1X_A, 4O7Q_A, 4QRL_A, 5CYB_A, 5JOE_A, 5ZGM_A, 6H8O_A.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.2c00069.

> Typical building block has a central segment of five residues and a few other segments around the central residue; local internal coordinate system of a building block around a central residue; comparison of the energies of native sequences and poly-X mutants; comparison of the RMSFs for dynamic trajectories for all the 18 proteins in the test sample; and distribution of the RMSD between the predicted structure using *TrRosetta* and the target structure (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Huan Yang − *Shanghai Institute for Advanced Immunochemical Studies, ShanghaiTech University, Shanghai 201210, China*; Email: lahplover@gmail.com

Francesco Zonta − *Shanghai Institute for Advanced Immunochemical Studies, ShanghaiTech University, Shanghai 201210, China*; orcid.org/0000-0001-8729-1071; Email: fzonta@shanghaitech.edu.cn

Author

Zhaoping Xiong − *Shanghai Institute for Advanced Immunochemical Studies, ShanghaiTech University, Shanghai 201210, China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.2c00069

Author Contributions

Conceptualization: H.Y., F.Z. ; Methodology: H.Y., Z.X.; Investigation: H.Y., Z.X., F.Z. ; Visualization: H.Y. ; Supervision: F.Z. ; Writing—original draft: H.Y., F.Z. ; Writing—review and editing: H.Y., F.Z.

Funding

This work was supported by the National Science Foundation of China (Grant No. 31770776 to FZ).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Dr. Michael Levitt for insightful discussion and for valuable suggestions and Dr. Sergio Pantano for helping us improve the manuscript after critically reading the first draft.

■ REFERENCES

(1) Levitt, M.; Warshel, A. Computer Simulation of Protein Folding. *Nature* **1975**, *253*, 694−698.

(2) Levitt, M. A Simplified Representation of Protein Conformations for Rapid Simulation of Protein Folding. *J. Mol. Biol.* **1976**, *104*, 59−107.

(3) Shaw, D. E.; Maragakis, P.; Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Eastwood, M. P.; Bank, J. A.; Jumper, J. M.; Salmon, J. K.; Shan, Y.; Wriggers, W. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **2010**, *330*, 341−346.

(4) Best, R. B.; Zhu, X.; Shim, J.; Lopes, P. E. M.; Mittal, J.; Feig, M.; MacKerell, A. D. Optimization of the Additive CHARMM All-Atom Protein Force Field Targeting Improved Sampling of the Backbone ϕ, ψ and Side-Chain χ_1 and χ_2 Dihedral Angles. *J. Chem. Theory Comput.* **2012**, *8*, 3257−3273.

(5) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. Ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from Ff99SB. *J. Chem. Theory Comput.* **2015**, *11*, 3696−3713.

(6) Tanaka, S.; Scheraga, H. A. Statistical Mechanical Treatment of Protein Conformation. III. Prediction of Protein Conformation Based on a Three-State Model. *Macromolecules* **1976**, *9*, 168−182.

(7) Miyazawa, S.; Jernigan, R. L. Estimation of Effective Interresidue Contact Energies from Protein Crystal Structures: Quasi-Chemical Approximation. *Macromolecules* **1985**, *18*, 534−552.

(8) Mirny, L. A.; Shakhnovich, E. I. How to Derive a Protein Folding Potential? A New Approach to an Old Problem. *J. Mol. Biol.* **1996**, *264*, 1164−1179.

(9) Simons, K. T.; Kooperberg, C.; Huang, E.; Baker, D. Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences Using Simulated Annealing and Bayesian Scoring Functions. *J. Mol. Biol.* **1997**, *268*, 209−225.

(10) Shen, M.; Sali, A. Statistical Potential for Assessment and Prediction of Protein Structures. *Protein Sci.* **2006**, *15*, 2507−2524.

(11) Alford, R. F.; Leaver-Fay, A.; Jeliazkov, J. R.; O'Meara, M. J.; DiMaio, F. P.; Park, H.; Shapovalov, M. V.; Renfrew, P. D.; Mulligan, V. K.; Kappel, K.; Labonte, J. W.; Pacella, M. S.; Bonneau, R.; Bradley, P.; Dunbrack, R. L.; Das, R.; Baker, D.; Kuhlman, B.; Kortemme, T.; Gray, J. J. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J. Chem. Theory Comput.* **2017**, *13*, 3031−3048.

(12) Davtyan, A.; Schafer, N. P.; Zheng, W.; Clementi, C.; Wolynes, P. G.; Papoian, G. A. AWSEM-MD: Protein Structure Prediction Using Coarse-Grained Physical Potentials and Bioinformatically Based Local Structure Biasing. *J. Phys. Chem. B* **2012**, *116*, 8494−8503.

(13) Warshel, A.; Levitt, M. Theoretical Studies of Enzymic Reactions: Dielectric, Electrostatic and Steric Stabilization of the Carbonium Ion in the Reaction of Lysozyme. *J. Mol. Biol.* **1976**, *103*, 227−249.

(14) Car, R.; Parrinello, M. Unified Approach for Molecular Dynamics and Density-Functional Theory. *Phys. Rev. Lett.* **1985**, *55*, 2471−2474.

(15) van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. A. ReaxFF: A Reactive Force Field for Hydrocarbons. *J. Phys. Chem. A* **2001**, *105*, 9396−9409.

(16) Zonta, F.; Mammano, F.; Torsello, M.; Fortunati, N.; Orian, L.; Polimeno, A. Role of Gamma Carboxylated Glu47 in Connexin 26 Hemichannel Regulation by Extracellular Ca2+: Insight from a Local Quantum Chemistry Study. *Biochem. Biophys. Res. Commun.* **2014**, *445*, 10−15.

(17) Bernardi, R. C.; Melo, M. C. R.; Schulten, K. Enhanced Sampling Techniques in Molecular Dynamics Simulations of Biological Systems. *Biochim. Biophys. Acta* **2015**, *1850*, 872−877.

(18) Souza, P. C. T.; Alessandri, R.; Barnoud, J.; Thallmair, S.; Faustino, I.; Grünewald, F.; Patmanidis, I.; Abdizadeh, H.; Bruininks, B. M. H.; Wassenaar, T. A.; Kroon, P. C.; Melcr, J.; Nieto, V.; Corradi, V.; Khan, H. M.; Domański, J.; Javanainen, M.; Martinez-Seara, H.; Reuter, N.; Best, R. B.; Vattulainen, I.; Monticelli, L.; Periole, X.; Tieleman, D. P.; de Vries, A. H.; Marrink, S. J. Martini 3: A General Purpose Force Field for Coarse-Grained Molecular Dynamics. *Nat. Methods* **2021**, *18*, 382−388.

(19) Machado, M. R.; Barrera, E. E.; Klein, F.; Sóñora, M.; Silva, S.; Pantano, S. The SIRAH 2.0 Force Field: Altius, Fortius, Citius. *J. Chem. Theory Comput.* **2019**, *15*, 2719−2733.

(20) Zonta, F.; Buratto, D.; Crispino, G.; Carrer, A.; Bruno, F.; Yang, G.; Mammano, F.; Pantano, S. Cues to Opening Mechanisms From in Silico Electric Field Excitation of Cx26 Hemichannel and in Vitro Mutagenesis Studies in HeLa Transfectans. *Front. Mol. Neurosci.* **2018**, *11*, 170.

(21) Orlandini, E.; Baiesi, M.; Zonta, F. How Local Flexibility Affects Knot Positioning in Ring Polymers. *Macromolecules* **2016**, *49*, 4656−4662.

(22) Gao, W.; Mahajan, S. P.; Sulam, J.; Gray, J. J. Deep Learning in Protein Structural Modeling and Design. *Patterns* **2020**, *1*, 100142.

(23) Kuhlman, B.; Bradley, P. Advances in Protein Structure Prediction and Design. *Nat. Rev. Mol. Cell Biol.* **2019**, *20*, 681−697.

(24) Wang, S.; Sun, S.; Li, Z.; Zhang, R.; Xu, J. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *PLoS Comput. Biol.* **2017**, *13*, No. e1005324.

(25) Yang, J.; Anishchenko, I.; Park, H.; Peng, Z.; Ovchinnikov, S.; Baker, D. Improved Protein Structure Prediction Using Predicted Interresidue Orientations. *Proc. Natl. Acad. Sci. U. S. A.* **2020**, *117*, 1496−1503.

(26) Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G. R.; Wang, J.; Cong, Q.; Kinch, L. N.; Schaeffer, R. D.; Millán, C.; Park, H.; Adams, C.; Glassman, C. R.; DeGiovanni, A.; Pereira, J. H.; Rodrigues, A. V.; van Dijk, A. A.; Ebrecht, A. C.; Opperman, D. J.; Sagmeister, T.; Buhlheller, C.; Pavkov-Keller, T.; Rathinaswamy, M. K.; Dalwadi, U.; Yip, C. K.; Burke, J. E.; Garcia, K. C.; Grishin, N. V.; Adams, P. D.; Read, R. J.; Baker, D. Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network. *Science* **2021**, *373*, 871−876.

(27) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.

(28) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified Rational Protein Engineering with Sequence-Based Deep Representation Learning. *Nat. Methods* **2019**, *16*, 1315−1322.

(29) Rao, R.; Liu, J.; Verkuil, R.; Meier, J.; Canny, J. F.; Abbeel, P.; Sercu, T.; Rives, A. MSA Transformer. *bioRxiv*, 2021, DOI: 10.1101/2021.02.12.430858.

(30) Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; Fergus, R. Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences. *Proc. Natl. Acad. Sci. U. S. A.* **2021**, *118*, No. e2016239118.

(31) Hermann, J.; Schätzle, Z.; Noé, F. Deep-Neural-Network Solution of the Electronic Schrödinger Equation. *Nat. Chem.* **2020**, *12*, 891−897.

(32) Pfau, D.; Spencer, J. S.; Matthews, A. G. D. G.; Foulkes, W. M. C. *Ab Initio* Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks. *Phys. Rev. Res.* **2020**, *2*, No. 033429.

(33) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(34) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1: An Extensible Neural Network Potential with DFT Accuracy at Force Field Computational Cost. *Chem. Sci.* **2017**, *8*, 3192−3203.

(35) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields. *Nat. Commun.* **2018**, *9*, 3887.

(36) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine Learning of Accurate Energy-Conserving Molecular Force Fields. *Sci. Adv.* **2017**, *3*, No. e1603015.

(37) Husic, B. E.; Charron, N. E.; Lemm, D.; Wang, J.; Pérez, A.; Majewski, M.; Krämer, A.; Chen, Y.; Olsson, S.; de Fabritiis, G.; Noé, F.; Clementi, C. Coarse Graining Molecular Dynamics with Graph Neural Networks. *J. Chem. Phys.* **2020**, *153*, 194101.

(38) Orengo, C. A.; Michie, A. D.; Jones, S.; Jones, D. T.; Swindells, M. B.; Thornton, J. M. CATH − a Hierarchic Classification of Protein Domain Structures. *Structure* **1997**, *5*, 1093−1109.

(39) Murzin, A. G.; Brenner, S. E.; Hubbard, T.; Chothia, C. SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures. *J. Mol. Biol.* **1995**, *247*, 536−540.

(40) Cossio, P.; Trovato, A.; Pietrucci, F.; Seno, F.; Maritan, A.; Laio, A. Exploring the Universe of Protein Structures beyond the Protein Data Bank. *PLoS Comput. Biol.* **2010**, *6*, No. e1000957.

(41) Mackenzie, C. O.; Zhou, J.; Grigoryan, G. Tertiary Alphabet for the Observable Protein Structural Universe. *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *113*, E7438−E7447.

(42) Nepomnyachiy, S.; Ben-Tal, N.; Kolodny, R. Complex Evolutionary Footprints Revealed in an Analysis of Reused Protein Segments of Diverse Lengths. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114*, 11703−11708.

(43) Kolodny, R.; Nepomnyachiy, S.; Tawfik, D. S.; Ben-Tal, N. Bridging Themes: Short Protein Segments Found in Different Architectures. *Mol. Biol. Evol.* **2021**, *38*, 2191−2208.

(44) Deng, H.; Jia, Y.; Zhang, Y. 3DRobot: Automated Generation of Diverse and Well-Packed Protein Structure Decoys. *Bioinformatics* **2016**, *32*, 378−387.

(45) Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. Improved Side-Chain Torsion Potentials for the Amber Ff99SB Protein Force Field. *Proteins: Struct., Funct., Bioinf.* **2010**, *78*, 1950−1958.

(46) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics. *PLoS Comput. Biol.* **2017**, *13*, No. e1005659.

(47) Hoang, T. X.; Trovato, A.; Seno, F.; Banavar, J. R.; Maritan, A. Geometry and Symmetry Presculpt the Free-Energy Landscape of Proteins. *Proc. Natl. Acad. Sci. U. S. A.* **2004**, *101*, 7960−7964.

(48) Trifonov, E. N. Consensus Temporal Order of Amino Acids and Evolution of the Triplet Code. *Gene* **2000**, *261*, 139−151.

(49) Wang, G.; Dunbrack, R. L. PISCES: A Protein Sequence Culling Server. *Bioinformatics* **2003**, *19*, 1589−1591.

(50) Steinegger, M.; Meier, M.; Mirdita, M.; Vöhringer, H.; Haunsberger, S. J.; Söding, J. HH-Suite3 for Fast Remote Homology Detection and Deep Protein Annotation. *BMC Bioinform.* **2019**, *20*, 473.

(51) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems December 2017*; 2017; pp. 6000−6010.

(52) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* January 29, 2017.