Patterns

CellPress
OPEN ACCESS

Review

Deep Learning in Protein Structural Modeling and Design

Wenhao Gao,[1] Sai Pooja Mahajan,[1] Jeremias Sulam,[2] and Jeffrey J. Gray[1,*]
[1]Department of Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, MD 21218, USA
[2]Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA
*Correspondence: jgray@jhu.edu

https://doi.org/10.1016/j.patter.2020.100142

THE BIGGER PICTURE Proteins are linear polymers that fold into an incredible variety of three-dimensional structures that enable sophisticated functionality for biology. Computational modeling allows scientists to predict the three-dimensional structure of proteins from genomes, predict properties or behavior of a protein, and even modify or design new proteins for a desired function. Advances in machine learning, especially deep learning, are catalyzing a revolution in the paradigm of scientific research. In this review, we summarize recent work in applying deep learning techniques to tackle problems in protein structural modeling and design. Some deep learning-based approaches, especially in structure prediction, now outperform conventional methods, often in combination with higher-resolution physical modeling. Challenges remain in experimental validation, benchmarking, leveraging known physics and interpreting models, and extending to other biomolecules and contexts.

1 2 3 4 5 **Development/Pre-production:** Data science output has been rolled out/validated across multiple domains/problems

SUMMARY

Deep learning is catalyzing a scientific revolution fueled by big data, accessible toolkits, and powerful computational resources, impacting many fields, including protein structural modeling. Protein structural modeling, such as predicting structure from amino acid sequence and evolutionary information, designing proteins toward desirable functionality, or predicting properties or behavior of a protein, is critical to understand and engineer biological systems at the molecular level. In this review, we summarize the recent advances in applying deep learning techniques to tackle problems in protein structural modeling and design. We dissect the emerging approaches using deep learning techniques for protein structural modeling and discuss advances and challenges that must be addressed. We argue for the central importance of structure, following the "sequence → structure → function" paradigm. This review is directed to help both computational biologists to gain familiarity with the deep learning methods applied in protein modeling, and computer scientists to gain perspective on the biologically meaningful problems that may benefit from deep learning techniques.

INTRODUCTION

Proteins are linear polymers that fold into various specific conformations to function. The incredible variety of three-dimensional (3D) structures determined by the combination and order in which 20 amino acids thread the protein polymer chain (sequence of the protein) enables the sophisticated functionality of proteins responsible for most biological activities. Hence, obtaining the structures of proteins is of paramount importance in both understanding the fundamental biology of health and disease and developing therapeutic molecules. While protein structure is primarily determined by sophisticated experimental techniques, such as X-ray crystallography,[1] NMR spectroscopy[2] and, increasingly, cryoelectron microscopy,[3] computational structure prediction from the genetically encoded amino acid sequence of a protein has been used as an alternative when experimental approaches are limited. Computational methods have been used to predict the structure of proteins,[4] illustrate the mechanism of biological processes,[5] and determine the properties of proteins.[6] Furthermore, all naturally occurring proteins are a result of an evolutionary process of random variants arising under various selective pressures. Through this process, nature has explored only a small subset of theoretically possible protein sequence space. To explore a broader sequence and

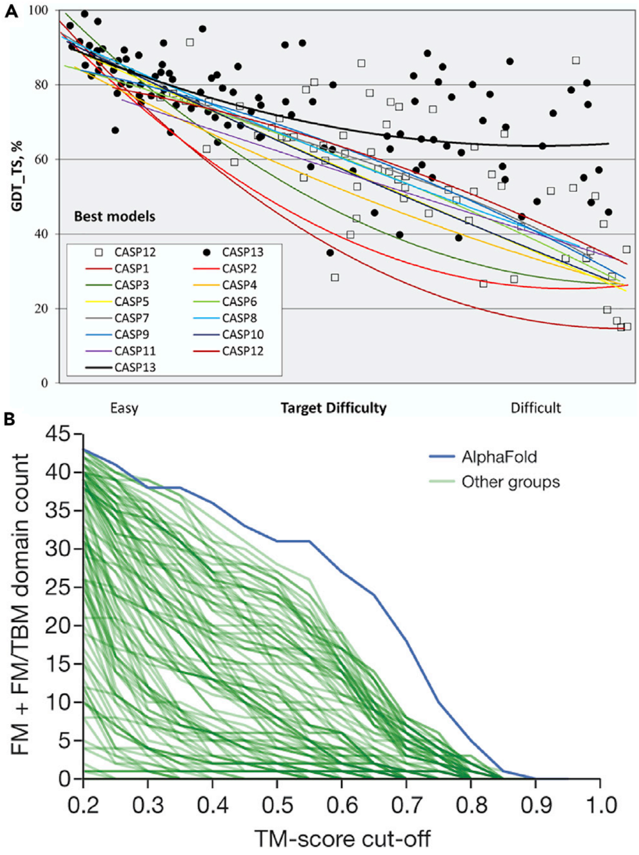

Figure 1. Striking Improvement in Model Accuracy in CASP13 Due to the Deployment of Deep Learning Methods

(A) Trend lines of backbone accuracy for the best models in each of the 13 CASP experiments. Individual target points are shown for the two most recent experiments. The accuracy metric, GDT_TS, is a multiscale indicator of the closeness of the Cα atoms in a model to those in the corresponding experimental structure (higher numbers are more accurate). Target difficulty is based on sequence and structure similarity to other proteins with known experimental structures (see Kryshtafovych et al.[4] for details). Figure from Kryshtafovych et al. (2019).[4]

(B) Number of FM + FM/TBM (FM, free modeling; TBM, template-based modeling) domains (out of 43) solved to a TM score threshold for all groups in CASP.[13] AlphaFold ranked first among them, showing that the progress is mainly due to the development of DL-based methods. Figure from Senior et al. (2020).[26]

structural space that potentially contains proteins with enhanced or novel properties, techniques, such as *de novo* design can be used to generate new biological molecules that have the potential to tackle many outstanding challenges in biomedicine and biotechnology.[7,8]

While the application of machine learning and more general statistical methods in protein modeling can be traced back decades,[9–13] recent advances in machine learning, especially in deep learning (DL)-related techniques,[14] have opened up new avenues in many areas of protein modeling.[15–18] DL is a set of machine learning techniques based on stacked neural network layers that parameterize functions in terms of compositions of affine transformations and non-linear activation functions. Their ability to extract domain-specific features that are adaptively learned from data for a particular task often enables them to surpass the performance of more traditional methods. DL has made dramatic impacts on digital applications like image classification,[19] speech recognition,[20] and game playing.[21] Success in these areas has inspired an increasing interest in more complex data types, including protein structures.[22] In the most recent Critical Assessment of Structure Prediction (CASP13 held in 2018),[4] a biennial community experiment to determine the state-of-the-art in protein structure prediction, DL-based methods accomplished a striking improvement in model accuracy (see Figure 1), especially in the "difficult" target category where comparative modeling (starting with a known, related structure) is ineffective. The CASP13 results show that the complex mapping from amino acid sequence to 3D protein structure can be successfully learned by a neural network and generalized to unseen cases. Concurrently, for the protein design problem, progress in the field of deep generative models has spawned a range of promising approaches.[23–25]

In this review, we summarize the recent progress in applying DL techniques to the problem of protein modeling and discuss the potential pros and cons. We limit our scope to protein structure and function prediction, protein design with DL (see Figure 2), and a wide array of popular frameworks used in these applications. We discuss the importance of protein representation, and summarize the approaches to protein design based on DL for the first time. We also emphasize the central importance of protein structure, following the sequence → structure → function paradigm and argue that approaches based on structures may be most fruitful. We refer the reader to other review papers for more information on applications of DL in biology and medicine,[16,15] bioinformatics,[27] structural biology,[17] folding and dynamics,[18,28] antibody modeling,[29] and structural annotation and prediction of proteins.[30,31] Because DL is a fast-moving, interdisciplinary field, we chose to include preprints in this review. We caution the reader that these contributions have not been peer-reviewed, yet are still worthy of attention now for their ideas. In fact, in communities such as computer science, it is not uncommon for manuscripts to remain in this stage indefinitely, and some seminal contributions, such as Kingma and Welling's definitive paper on autoencoders (AEs),[32] are only available as preprints. In addition, we urge caution with any protein design studies that are purely *in silico*, and we highlight those that include experimental validation as a sign of their trustworthiness.

PROTEIN STRUCTURE PREDICTION AND DESIGN

Problem Definition

The prediction of protein 3D structure from amino acid sequence has been a grand challenge in computational biophysics for decades.[33,34] Folding of peptide chains is a fundamental concept in biophysics, and atomic-level structures of proteins and complexes are often the starting point to understand their function and to modulate or engineer them. Thanks to the recent advances in next-generation sequencing technology, there are now over 180 million protein sequences recorded in the UniProt dataset.[35] In contrast, only 158,000 experimentally determined structures are available in the Protein Data Bank. Thus, computational structure prediction is a critical problem of both practical and theoretical interest.

More recently, the advances in structure prediction have led to an increasing interest in the protein design problem. In design,

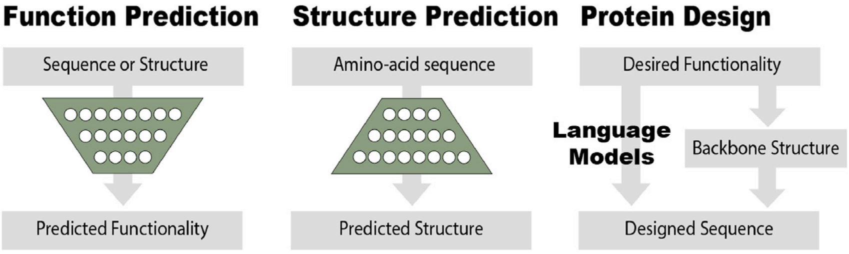

Figure 2. Schematic Comparison of Three Major Tasks in Protein Modeling: Function Prediction, Structure Prediction, and Protein Design

In function prediction, the sequence and/or the structure is known and the functionality is needed as output of a neural net. In structure prediction, sequence is known input and structure is unknown output. Protein design starts from desired functionality, or a step further, structure that can perform this functionality. The desired output is a sequence that can fold into the structure or has such functionality.

the objective is to obtain a novel protein sequence that will fold into a desired structure or perform a specific function, such as catalysis. Naturally occurring proteins represent only an infinitesimal subset of all possible amino acid sequences selected by the evolutionary process to perform a specific biological function.[7] Proteins with more robustness (higher thermal stability, resistance to degradation) or enhanced properties (faster catalysis, tighter binding) might lie in the space that has not been explored by nature, but is potentially accessible by *de novo* design. The current approach for computational *de novo* design is based on physical and evolutionary principles and requires significant domain expertise. Some successful examples include novel folds,[36] enzymes,[37] vaccines,[38] novel protein assemblies,[39] ligand-binding protein,[40] and membrane proteins.[41] While some papers occasionally refer to redesign of naturally occurring proteins or interfaces as "*de novo*", in this review we restrict that term only to works where completely new folds or interfaces are created.

Conventional Computational Approaches

The current methodology for computational protein structure prediction is largely based on Anfinsen's[42] thermodynamic hypothesis, which states that the native structure of a protein must be the one with the lowest free energy, governed by the energy landscape of all possible conformations associated with its sequence. Finding the lowest-energy state is challenging because of the immense space of possible conformations available to a protein, also known as the "sampling problem" or Levinthal's[43] paradox. Furthermore, the approach requires accurate free energy functions to describe the protein energy landscape and rank different conformations based on their energy, referred to as the "scoring problem." In light of these challenges, current computational techniques rely heavily on multiscale approaches. Low-resolution, coarse-grained energy functions are used to capture large-scale conformational sampling, such as the hydrophobic burial and formation of local secondary structural elements. Higher-resolution energy functions are used to explicitly model finer details, such as amino acid side-chain packing, hydrogen bonding, and salt bridges.[44]

Protein design problems, sometimes known as the inverse of structure prediction problems, require a similar toolbox. Instead of sampling the conformational space, a protein design protocol samples the sequence space that folds into the desired topology. Past efforts can be broadly divided into two broad classes: modifying an existing protein with known sequence and properties, or generating novel proteins with sequences and/or folds unrelated to those found in nature. The former class evolves an existing protein's amino acid sequence (and as a result, structure and properties) and can be loosely referred to as protein engineering or protein redesign. The latter class of methods is called *de novo* protein design, a term originally coined in 1997 when Dahiyat and Mayo[45] designed the FSD-1 protein, a soluble protein with a completely new sequence that folded into the previously known structure of a zinc finger. Korendovych and DeGrado's[46] recent retrospective chronicles the development of *de novo* design. Originally *de novo* design meant creation of entirely new proteins from scratch exploiting a target structure but, especially in the DL era, many authors now use the term to include methods that ignore structure in creating new sequences, often using extensive training data from known proteins in a particular functional class. In this review, we split our discussion of methods according to whether they trained directly between sequence and function (as certain natural language processing [NLP]-based DL paradigms allow), or whether they directly include protein structural data (like historical methods in rational protein design; see below in the section on "Protein Design").

Despite significant progress in the last several decades in the field of computational protein structure prediction and design,[7,34] accurate structure prediction and reliable design both remain challenging. Conventional approaches rely heavily on the accuracy of the energy functions to describe protein physics and the efficiency of sampling algorithms to explore the immense protein sequence and structure space. Both protein engineering and *de novo* approaches are often combined with experimental directed evolution[8,47] to achieve the optimal final molecules.[7]

DL ARCHITECTURES

In conventional computational approaches, predictions from data are made by means of physical equations and modeling. Machine learning puts forward a different paradigm in which algorithms automatically infer—or *learn*—a relationship between inputs and outputs from a set of hypotheses. Consider a collection of N training samples comprising features $\boldsymbol{x}$ in an input space $\mathcal{X}$ (e.g., amino acid sequences), and corresponding labels y in some output space $\mathcal{Y}$ (e.g., residue pairwise distances), where $\{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ are sampled independently and identically distributed from some joint distribution $\mathcal{P}$. In addition, consider a function $f : \mathcal{X} \to \mathcal{Y}$ in some function class $\mathcal{H}$, and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ that measures how much $f(\boldsymbol{x})$ deviates from the corresponding label y. The goal of supervised learning is to find a function $f \in \mathcal{H}$ that minimizes the expected loss, $\mathbb{E}[\ell(f(\boldsymbol{x}), y)]$, for $(\boldsymbol{x}, y)$ sampled from $\mathcal{P}$. Since one does not have access to the true distribution but rather N samples from it, the

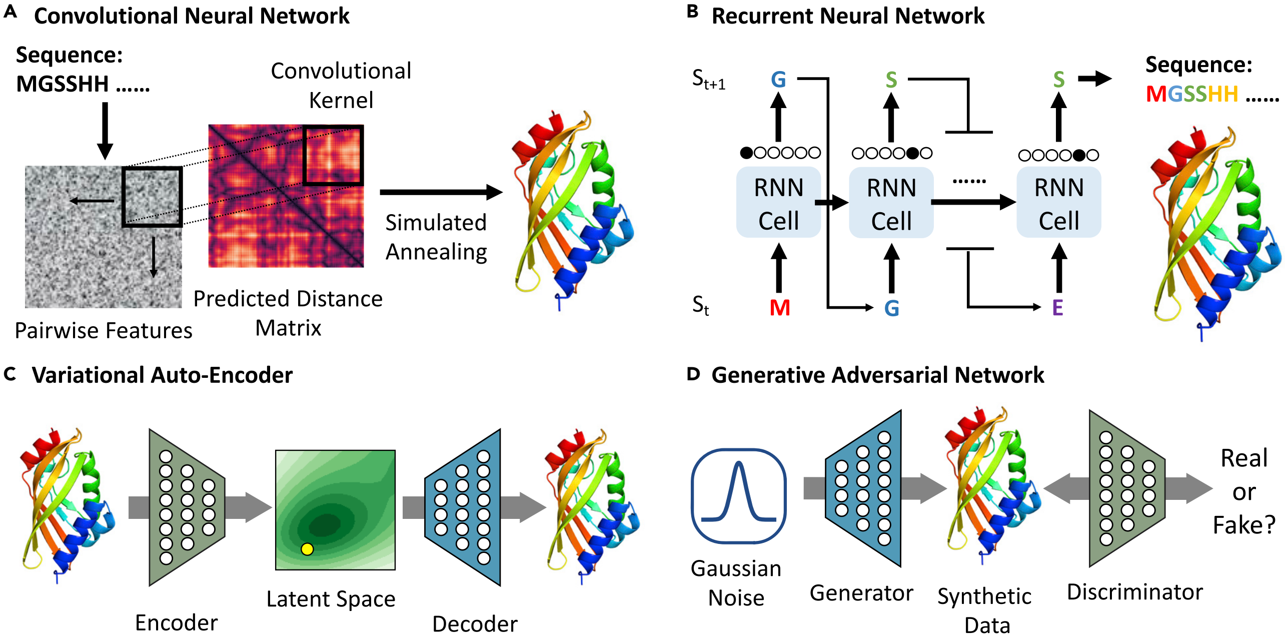

Figure 3. Schematic Representation of Several Architectures Used in Protein Modeling and Design

(A) CNNs are widely used in structure prediction.

(B) RNNs learn in an auto-regressive way and can be used for sequence generation.

(C) The VAE can be jointly trained by protein and properties to construct a latent space correlated with properties.

(D) In the GAN setting, a mapping from *a priori* distribution to the design space can be obtained via the adversarial training.

popular empirical risk minimization (ERM) approach seeks to minimize the loss over the training samples instead. In neural network models, in particular, the function class is parameterized by a collection of weights. Denoting these parameters collectively by $\boldsymbol{\theta}$, ERM boils down to an optimization problem of the form

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \ell(f_{\theta}(\boldsymbol{x}_i), y_i). \qquad \text{(Equation 1)}$$

The choice of the network determines how the hypothesis class is parameterized. Deep neural networks typically implement a non-linear function as the composition of affine maps, $\mathcal{W}_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_{l+1}}$, where $\mathcal{W}_l\boldsymbol{x} = W_l\boldsymbol{x} + \boldsymbol{b}_l$, and other non-linear activation functions, $\sigma(\bullet)$. Rectifying linear units and max-pooling are some of the most popular non-linear transformations applied in practice. The architecture of the model determines how these functions are composed, the most popular option being their sequential composition $f(\boldsymbol{x}) = \mathcal{W}_L\sigma(\mathcal{W}_{L-1}\sigma(\mathcal{W}_{L-2}\sigma(\ldots\mathcal{W}_2\sigma(\mathcal{W}_1\boldsymbol{x}))))$ for a network with L layers. Computing $f(\boldsymbol{x})$ is typically referred to as the forward pass.

We will not dwell on the details of the optimization problem in Equation (1), which is typically carried out via stochastic gradient descent algorithms or variations thereof, efficiently implemented via back-propagation (see instead, e.g., LeCun et al.,[14] Sun,[48] and Schmidhuber).[49] Rather, in this section we summarize some of the most popular models widely used in protein structural modeling, including how different approaches are best suited for particular data types or applications. High-level diagrams of the major architectures are shown in Figures 3.

Convolutional Neural Networks

Convolutional networks architectures[50] are most commonly applied to image analysis or other problems where shift-invariance or covariance is needed. Inspired by the fact that an object on an image can be shifted in the image and still be the same object, convolutional neural networks (CNNs) adopt convolutional kernels for the layer-wise affine transformation to capture this translational invariance. A 2D convolutional kernel $\mathbf{w}$ applied to a 2D image data $\boldsymbol{x}$ can be defined as

$$\boldsymbol{S}(i,j) = (\boldsymbol{x} * \mathbf{w})(i,j) = \sum_m \sum_n \boldsymbol{x}(m,n)\mathbf{w}(i-m, j-n), \qquad \text{(Equation 2)}$$

where $\boldsymbol{S}(i,j)$ represents the output at position (i,j), $\boldsymbol{x}(m,n)$ is the value of the input $\boldsymbol{x}$ at position (m,n), $\mathbf{w}(i-m, j-n)$ is the parameter of kernel $\mathbf{w}$ at position $(i-m, j-n)$, and the summation is over all possible positions. An important variant of CNN is the residual network (ResNet),[51] which incorporates skip-connections between layers. These modification have shown great advantages in practice, aiding the optimization of these typically huge models. CNNs, especially ResNets, have been widely used in protein structure prediction. An example is AlphaFold,[22] which used ResNets to predict protein inter-residue distance maps from amino acid sequences (Figure 3A).

Recurrent Neural Networks

Recurrent architectures are based on applying several iterations of the same function along a sequential input.[52] This can be seen as an *unfolded* architecture, and it has been widely used to process sequential data, such as time series data and written text

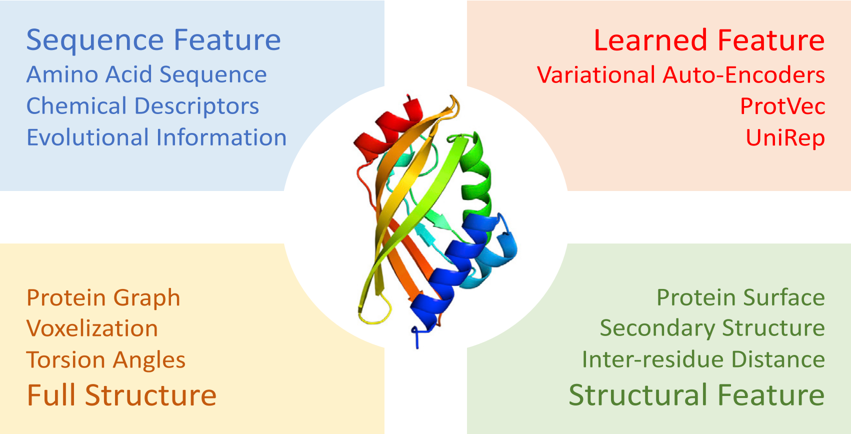

Figure 4. Different Types of Representation Schemes Applied to a Protein

(i.e., NLP). With an initial hidden state $\boldsymbol{h}^{(0)}$ and sequential data $[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}]$, we can obtain hidden states recursively:

$$\boldsymbol{h}^{(t)} = g^{(t)}\left(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(t-1)}, \boldsymbol{x}^{(t-2)}, \ldots, \boldsymbol{x}^{(1)}\right) = f\left(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta}\right), \tag{Equation 3}$$

where f represents a function or transformation from one position to the next, and $g^{(t)}$ represents the accumulative transformation up to position t. The hidden state vector at position i, $\boldsymbol{h}^{(i)}$, contains all the information that has been seen before. As the same set of parameters (usually called a cell) can be applied recurrently along the sequential data, an input of variable length can be fed to a recurrent neural network (RNN). Due to the gradient vanishing and explosion problem (the error signal decreases or increases exponentially during training), more recent variants of standard RNNs, namely long short-term memory (LSTM)[53] and gated recurrent unit[54] are more widely used. An example of an RNN approach in the context of protein structure prediction is using an N-terminal subsequence of a protein to predict the next amino acid in the protein sequence (Figure 3B; e.g., Müller et al.[55]).

In conjunction with recurrent networks, attention mechanisms were first proposed (in an encoder-decoder framework) to learn which parts of a source sentence are most relevant to predicting a target word.[56] Compared with RNN models, attention-based models are more parallelizable and better at capturing long-range dependencies, and they are driving big advances in NLP.[57,58] Recently, the transformer model, which solely adopted attention layers without any recurrent or convolutional layers, was able to surpass state-of-the-art methods on language translation tasks.[57] For proteins, these methods could learn which parts of an amino acid sequence are critical to predicting a target residue or the properties of a target residue. For example, transformer-based models have been used to generate protein sequences conditioned on target structure,[23] learn protein sequence data to predict secondary structure and fitness landscapes,[59] and to encode the context of the binding partner in antibody-antigen binding surface prediction.[60]

Variational Autoencoder

AEs,[61] unlike the networks discussed so far, provide a model for unsupervised learning. Within this unsupervised framework, an AE does not learn labeled outputs but instead attempts to learn some representation of the original input. This is typically accomplished by training two parametric maps: an encoder function $g: \mathcal{X} \rightarrow \mathbb{R}^m$ that maps an input $\boldsymbol{x}$ to an m-dimensional representation or latent space, and a decoder intended to implement the inverse map so that $f(g(\boldsymbol{x})) \approx \boldsymbol{x}$. Typically, the latent representation is of small dimension (m is smaller than the ambient dimension of $\mathcal{X}$) or constrained in some other way (e.g., through sparsity).

Variational autoencoders (VAEs),[32,62] in particular, provide a stochastic map between the input space and the latent space. This map is beneficial because, while the input space may have a highly complex distribution, the distribution of the representation $\boldsymbol{z}$ can be much simpler; e.g., Gaussian. These methods derive from variational inference, a method from machine learning that approximates probability densities through optimization.[63] The stochastic encoder, given by the *inference model* $q_\varphi(\boldsymbol{z}|\boldsymbol{x})$ and parametrized by weights φ, is trained to approximate the true posterior distribution of the representation given the data, $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. The decoder, on the other hand, provides an estimate for the data given the representation, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. Direct optimization of the resulting objective is intractable, however. Thus, training is done by maximizing the "evidence lower bound," $\mathcal{L}_{\boldsymbol{\theta},\varphi}(\boldsymbol{x})$, instead, which provides a lower bound on the log-likelihood of the data:

$$\mathcal{L}_{\boldsymbol{\theta},\varphi}(\boldsymbol{x}) = \mathrm{E}_{\boldsymbol{z}\sim q_\varphi(\boldsymbol{z}|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) - D_{KL}(q_\varphi(\boldsymbol{z}|\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})). \tag{Equation 4}$$

Here, $D_{KL}(q_\varphi || p_{\boldsymbol{\theta}})$ is the Kullback-Leibler divergence, which quantifies the distance between distributions q_φ and $p_{\boldsymbol{\theta}}$. Employing Gaussians for the factorized variational and likelihood distributions, as well as using a change of variables via differentiable maps, allows for the efficient optimization of these architectures.

An example of applying VAE in the protein modeling field is learning a representation of antimicrobial protein sequences (Figure 3C; e.g., Das et al.[64]). The resulting continuous real-valued representation can then be used to generate new sequences likely to have antimicrobial properties.

Generative Adversarial Network

Generative adversarial networks (GANs)[65] are another class of unsupervised (generative) models. Unlike VAEs, GANs are trained by an adversarial game between two models, or networks: a *generator*, G, which given a sample, $\boldsymbol{z}$, from some simple distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$ (e.g., Gaussian), seeks to map it to the distribution of some data class (e.g., naturally looking images); and a *discriminator*, D, whose task is to detect whether the images are real (i.e., belonging to the true distribution of the data,

Table 1. Features Contained by CUProtein Dataset

Feature Name	Description	Dimensions	Type	IO
AA Sequence	sequence of amino acid	n × 1	21 chars	input
PSSM	position-specific scoring matrix, a residue-wise score for motifs appearance	n × 21	real [0, 1]	input
MSA covariance	covariance matrix across homologous NA sequences	n × n	real [0, 1]	input
SS	a coarse categorized secondary structure (Q3 or Q8)	n × 1	3 or 8 chars	input
Distance matrices	pairwise distance between residues (C_α or C_β)	n × n	positive real (Å)	output
Torsion angles	variable dihedral angles for each residues (φ, ψ)	n × 2	real $[-\pi, +\pi]$ (radians)	output

n, number of residues in one protein. Data from Drori et al.[78]

$p_{\text{data}}(\boldsymbol{x})$), or fake (produced by the generator). With this game-based setup, the generator model is trained by maximizing the error rate of the discriminator, thereby training it to "fool" the discriminator. The discriminator, on the other hand, is trained to foil such fooling. The original objective function as formulated by Goodfellow et al.[65] is:

$$\min_G \max_D V(D,G) = \mathrm{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathrm{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]. \qquad \text{(Equation 5)}$$

Training is performed by stochastic optimization of this differentiable loss function. While intuitive, this original GAN objective can suffer from issues, such as mode collapse and instabilities during training. The Wasserstein GAN (WGAN)[66] is a popular extension of GAN which introduces a Wasserstein-1 distance measure between distributions, leading to easier and more robust training.[67]

An example of a GAN, in the context of protein modeling is learning the distribution of protein backbone distances to generate novel protein-like folds (Figure 3D).[68] During training, one network G generates folds, and a second network D aims to distinguish between generated folds and fake folds.

PROTEIN REPRESENTATION AND FUNCTION PREDICTION

One of the most fundamental challenges in protein modeling is the prediction of functionality from sequence or structure. Function prediction is typically formulated as a supervised learning problem. The property to predict can either be a protein-level property, such as a classification as an enzyme or non-enzyme,[69] or a residue-level property, such as the sites or motifs of phosphorylation (DeepPho)[70] and cleavage by proteases.[71] The challenging part here and in the following models is how to represent the protein. Representation refers to the encoding of a protein that serves as an input for prediction tasks or the output for generation tasks. Although a deep neural network is in principle capable of extracting complex features, a well-chosen representation can make learning more effective and efficient.[72] In this section, we will introduce the commonly used representations of proteins in DL models (Figure 4): sequence-based, structure-based, and one special form of representation relevant to computational modeling of proteins: coarse-grained models.

Amino Acid Sequence as Representation

As the amino acid sequence contains the information essential to reach the folded structure for most proteins,[42] it is widely used as an input in functional prediction and structure prediction tasks. The amino acid sequence, like other sequential data, is typically converted into one-hot encoding-based representation (each residue is represented with one high bit to identify the amino acid type and all the others low) that can be directly used in many sequence-based DL techniques.[73,74] However, this representation is inherently sparse and, thus, sample-inefficient. There are many easily accessible additional features that can be concatenated with amino acid sequences providing structural, evolutionary, and biophysical information. Some widely used features include predicted secondary structure, high-level biological features, such as sub-cellular localization and unique functions,[75] and physical descriptors, such as AAIndex,[76] hydrophobicity, ability to form hydrogen bonds, charge, solvent-accessible surface area, etc. A sequence can be augmented with additional data from sequence databases, such as multiple sequence alignments (MSA) or position-specific scoring matrices (PSSMs),[77] or pairwise residue co-evolution features. Table 1 lists typical features as used in CUProtein.[78]

Learned Representation from Amino Acid Sequence

Because the performance of machine learning algorithms highly depends on the features we choose, labor-intensive and domain-based feature engineering was vital for traditional machine learning projects. Now, the exceptional feature extraction ability of neural networks makes it possible to "learn" the representation, with or without giving the model any labels.[72] As publicly available sequence data are abundant (see Table 2), a well-learned representation that utilizes these data to capture more information is of particular interest. The class of algorithms that address the label-less learning problem fall under the umbrella of unsupervised or semi-supervised learning, which extracts information from unlabeled data to reduce the number of labeled samples needed.

Table 2. A Summary of Publicly Available Molecular Biology Databases

Dataset	Description	N	Website
European Bioinformatics Institute (EMBL-EBI)	a collections of wide range of datasets	–	https://www.ebi.ac.uk
National Center for Biotechnology Information (NCBI)	a collections of biomedical and genomic databases	–	https://www.ncbi.nlm.nih.gov
Protein Data Bank (PDB)	3D structural data of biomolecules, such as proteins and nucleic acids	∼160,000	https://www.rcsb.org
Nucleic Acid Database (NDB)	structure of nucleic acids and complex assemblies	∼10,560	http://ndbserver.rutgers.edu
Universal Protein Resource (UniProt)	protein sequence and function infromations	$\sim 562,000$	http://www.uniprot.org/
Sequence Read Archive (SRA)	raw sequence data from "next-generation" sequencing technologies	$\sim 3 \times 10^{16}$	NCBI database

The most straightforward way to learn from amino acid sequence is to directly apply NLP algorithms. Word2Vec[79] and Doc2Vec[80] are groups of algorithms widely used for learning word or paragraph embeddings. These models are trained by either predicting a word from its context or predicting its context from one central word. To apply these algorithm, Asgari and Mofrad[81] first proposed a Word2Vec-based model called BioVec, which interprets the non-overlapping 3-mer sequence of amino acids (e.g., alanine-glutamine-lysine or AQL) as "words" and lists of shifted "words" as "sentences." They then represent a protein as the summation of all overlapping sequence fragments of length *k*, or *k*-mers (called ProtVec). Predictions based on the ProtVec representation outperformed state-of-the-art machine learning methods in the Pfam protein family[82] classification (93% accuracy for ~7,000 proteins, versus 69.1%–99.6%[83] and 75%[84] for previous methods). Many Doc2Vec-type extensions were developed based on the 3-mer protocol. Yu et al.[85] showed that non-overlapping *k*-mers perform better than the overlapping ones, and Yang et al.[86] compared the performance of all Doc2Vec frameworks for thermostability and enantioselectivity prediction.

In these approaches, the three-residue segmentation of a protein sequence is arbitrary and does not embody any biophysical meaning. Alternatively, Alley et al.[87] directly used an RNN (unidirectional multiplicative long-short-term-memory or mLSTM)[88] model, called UniRep, to summarize arbitrary length protein sequences into a fixed-length real representation by averaging over the representation of each residue.[87] Their representation achieved lower mean squared errors on 15 property prediction tasks (e.g., absorbance, activity, stability) compared with former models, including Yang et al.'s[86] Doc2Vec. Heinzinger et al.[89] adopted the bidirectional LSTM in a manner similar to Peters et al.'s[90] ELMo (Embeddings from Language Models) model and surpassed Asgari and Mofrad's[81] Word2Vec model at predicting secondary structure and regions with intrinsic disorder at the per-residue level.[89] The success of the transformer model in language processing, especially those trained on large number of parameters, such as BERT[58] and GPT3,[91] has inspired its application in biological sequence modeling. Rives et al.[59] trained a transformer model with 670 million parameters on 86 billion amino acids across 250 million protein sequences spanning evolutionary diversity. Their transformer model was superior to traditional LSTM-based models on tasks, such as the prediction of secondary structure and long-range contacts, as well as the effect of mutations on activity on deep mutational scanning benchmarks.

AEs can also provide representations for subsequent supervised tasks.[32] Ding et al.[92] showed that a VAE model is able to capture evolutionary relationships between sequences and stability of proteins, while Sinai et al.[93] and Riesselman et al.[94] showed that the latent vectors learned from VAEs are able to predict the effects of mutations on fitness and activity for a range of proteins, such as poly(A)-binding protein, DNA methyltransferase, and β-lactamase. Recently, a lower-dimensional embedding of the sequence was learned for the more complex task of structure prediction.[78] Alley et al.'s[87] UniRep surpassed former models, but since UniRep is trained on 24 million sequences and previous models (e.g., Prot2Vec) were trained on much smaller datasets (0.5 million), it is not clear if the improvement was due to better methods or the larger training dataset. Rao et al.[95] introduced multiple biological-relevant semi-supervised learning tasks, TAPE, and benchmarked the performance against various protein representations. Their results show conventional alignment-based inputs still outperform current self-supervised models on multiple tasks, and the performance on a single task cannot evaluate the capacity of models. A comprehensive and persuasive comparison of representations is required.

Structure as Representation

Since the most important functions of a protein (e.g., binding, signaling, catalysis) can be traced back to the 3D structure of the protein, direct use of 3D structural information, and analogously, learning a good representation based on 3D structure, are highly desired. The direct use of raw 3D representations (such as coordinates of atoms) is hindered by considerable challenges, including the processing of unnecessary information due to translation, rotation, and permutation of atomic indexing. Townshend et al.[96,97] and Simonovsky and Meyers[96,97] obtained a translationally invariant, 3D representation of each residue by voxelizing its atomic neighborhood for a grid-based 3D CNN model. The work of Kolodny et al.,[98] Taylor,[99] and Li and

Koehl[100] representing the 3D structure of a protein as 1D strings of geometric fragments for structure comparison and fold recognition may also prove useful in DL approaches. Alternatively, the torsion angles of the protein backbone, which are invariant to translation and rotation, can fully recapitulate protein backbone structure under the common assumption that variation in bond lengths and angles is negligible. AlQuraishi[101] used backbone torsion angles to represent the 3D structure of the protein as a 1D data vector. However, because a change in a backbone torsion angle at a residue affects the inter-residue distances between all preceding and subsequent residues, these 1D variables are highly interdependent, which can frustrate learning. To circumvent these limitations, many approaches use 2D projections of 3D protein structure data, such as residue-residue distance and contact maps,[24,102] and pseudo-torsion angles and bond angles that capture the relative orientations between pairs of residues.[103] While these representations guarantee translational and rotational invariance, they do not guarantee invertibility back to the 3D structure. The structure must be reconstructed by applying constraints on distance or contact parameters using algorithms, such as gradient descent minimization, multidimensional scaling, a program like the Crystallography and NMR system (CNS),[104] or in conjunction with an energy-function-based protein structure prediction program.[22]

An alternative to the above approaches for representing protein structures is the use of a graph, i.e., a collection of nodes or vertices connected by edges. Such a representation is highly amenable to the graph neural network (GNN) paradigm,[105] which has recently emerged as a powerful framework for non-Euclidean data[106] in which the data are represented with relationships and inter-dependencies, or edges between objects or nodes.[107] While the representation of proteins as graphs and the application of graph theory to study their structure and properties has a long history,[108] the efforts to apply GNNs to protein modeling and design is quite recent. As a benchmark, many GNNs[69,109] have been applied to classify enzymes from non-enzymes in the PROTEINS[110] and D&D[111] datasets. Fout et al.[112] utilized a GNN in developing a model for protein-protein interface prediction. In their model, the node feature comprised residue composition and conservation, accessible surface area, residue depth, and protrusion index; and the edge feature comprised a distance and an angle between the normal vectors of the amide plane of each node/residue. A similar framework was used to predict antibody-antigen binding interfaces.[60] Zamora-Resendiz and Crivelli[113] and Gligorijevic et al.[114] further generalized and validated the use of graph-based representations and the graph convolutional network (GCN) framework in protein function prediction tasks, using a class activation map to interpret the structural determinants of the functionalities. Torng and Altman[115] applied GCNs to model pocket-like cavities in proteins to predict the interaction of proteins with small molecules, and Ingraham et al.[23] adopted a graph-based transformer model to perform a protein sequence design task. These examples demonstrate the generality and potential of the graph-based representation and GNNs to encode structural information for protein modeling.

The surface of the protein or a cavity is an information-rich region that encodes how a protein may interact with other molecules and its environment. Recently, Gainza et al.[116] used a geometric DL framework[117] to learn a surface-based representation of the protein, called MaSIF. They calculated "fingerprints" for patches on the protein surface using geodesic convolutional layers, which were further used to perform tasks, such as binding site prediction or ultra-fast protein-protein interaction (PPI) search. The performance of MaSIF approached the baseline of current methods in docking and function prediction, providing a proof-of-concept to inspire more applications of geometry-based representation learning.

Score Function and Force Field

A high-quality force field (or, more generally, score function) for sampling and/or ranking models (decoys) is one of the most vital requirements for protein structural modeling.[118] A force field describes the potential energy surface of a protein. A score function may contain knowledge-based terms that do not necessarily have a valid physical meaning, and they are designed to distinguish near-native conformations from non-native ones (for example, learning the GDT_TS).[119] A molecular dynamics (MD) or Monte Carlo (MC) simulation with a state-of-the-art force field or score function can reproduce reasonable statistical behaviors of biomolecules.[120–122]

Current DL-based efforts to learn the force field can be divided into two classes: "fingerprint" based and graph based. Behler and Parrinello[123] developed roto-translationally invariant features, i.e., the Behler-Parrinello fingerprint, to encode the atomic environment for neural networks to learn potential surfaces from density functional theory (DFT) calculations. Smith et al. extended this framework and tested its accuracy by simulating systems up to 312 atoms (Trp-cage) for 1 ns.[124,125] Another family that includes deep tensor neural networks[126] and SchNet[127] uses graph convolutions to learn a representation for each atom within its chemical environment. Although the prediction quality and the ability to learn a representation with novel chemical insight make the graph-based approach increasingly popular,[28] the application scales poorly to larger systems and thus has mainly focused on small organic molecules.

We anticipate a shift toward DL-based score functions because of the enormous gains in speed and efficiency. For example, Zhang et al.[128] showed that MD simulation on a neural potential was able to reproduce energies, forces, and time-averaged properties comparable with *ab initio* MD (AIMD) at a cost that scales linearly with system size, compared with cubic scaling typical for AIMD with DFT. Although these force fields are, in principle, generalizable to larger systems, direct applications of learned potentials to model full proteins are still rare. PhysNet, trained on a set of small peptide fragments (at most eight heavy atoms), was able to generalize to deca-alanine (Ala10),[129] and ANI-1x and AIMNet have been tested on chignolin (10 residues) and Trp-cage (20 residues) within the ANI-MD benchmark dataset.[125,130] Lahey and Rowley[131] and Wang et al.[132] combined the quantum mechanics/molecular mechanics (QM/MM) strategy[133] and the neural potential to model docking with small ligands and larger proteins.[131,132] Recently, Wang et al.[134] proposed an end-to-end differential MM force field by training a GNN on energies and forces to learn atom-typing and force field parameters.

Coarse-Grained Models

Coarse-grained models are higher-level abstractions of biomolecules, such as using a single pseudo-atom or a bead to represent multiple atoms, grouped based on local connectivity and/or chemical properties. Coarse graining smoothens out the energy landscape, and thereby helps avoid trapping in local minima and speeds up conformational sampling.[135] One can learn the atomic-level properties to construct a fast and accurate neural coarse-grained model once the coarse-grained mapping is given. Early attempts to apply DL-based methods to coarse-graining focus on water molecules with the roto-translationally invariant features.[136,137] Wang et al.[138] developed CGNet and learned the coarse-grained model of the mini protein, chignolin, in which the atoms of a residue are mapped to the corresponding C_α atom. The free energy surface learned with CGNet is quantitatively correct and MD simulations performed with CGNet potentially predict the same set of metastable states (folded, unfolded, and misfolded). Another critical question for coarse graining is determining which sets of atoms to map into a united atom. For example, one choice is to use a single coarse-grained atom to represent a whole residue, and a different choice is to use two coarse-grained atoms, one to represent the backbone and the other to represent the side chain. To determine the optimal choice, Wang and Gómez-Bombarelli[139] applied an encoder-decoder-based model to explicitly learn the lower-dimensional representation of proteins by minimizing the information loss at different levels of coarse graining. Li et al.[140] treated this problem as a graph segmentation problem and presented a GNN-based coarse-graining mapping predictor called Deep Supervised Graph Partitioning Model.

STRUCTURE DETERMINATION

The most successful application of DL in the field of protein modeling so far has been the prediction of protein structure. Protein structure prediction is formulated as a well-defined problem with clear inputs and outputs: predict the 3D structure (output) given amino acid sequences (input), with the experimental structures as the ground truth (labels). This problem perfectly fits the classical supervised learning approach, and once the problem is defined in these terms, the remaining challenge is to choose a framework to handle the complex relationship between input and output. The CASP experiment for structure prediction is held every 2 years and served as a platform for DL to compete with state-of-the-art methods and, impressively, outshine them in certain categories. We will first discuss the application of DL to the protein folding problem, and then comment on some problems related to structure determination. Table 3 summarizes major DL efforts in structure prediction.

Protein Structure Prediction

Before the notable success of DL at CASP12 (2016) and CASP13 (2018), the state-of-the-art methodology used complex workflows based on a combination of fragment insertion and structure optimization methods, such as simulated annealing with a score function or energy potential. Over the last decade, the introduction of co-evolution information in the form of evolutionary coupling analysis (ECA)[154] improved predictions. ECA relies on the rationale that residue pairs in contact in 3D space tend to evolve or mutate together; otherwise, they would disrupt the structure to destabilize the fold or render a large conformational change. Thus, evolutionary couplings from sequencing data suggest distance relationships between residue pairs and aid structure construction from sequence through contact or distance constraints. Because co-evolution information relies on statistical averaging of sequence information from a large number of MSAs,[145,155,156] this approach is not effective when the protein target has only a few sequence homologs. Neural networks were, at first, introduced to deduce evolutionary couplings between distant homologs, thereby improving ECA-type contact predictions for contact-assisted protein folding.[154] While the application of neural networks to learn inter-residue protein contacts dates back to the early 2000s,[157,158] more recently this approach was adopted by MetaPSICOV (two-layer NN),[146] PConsC2 (two-layer NN),[145] and CoinDCA-NN (five-layer NN),[155] which combined neural networks with ECAs. However, there was no significant advantage to neural networks compared with other machine learning methods at that time.[159]

In 2017, Wang et al.[102] proposed RaptorX-Contact, a residual neural network (ResNet)-based model,[51] which, for the first time used a *deep* neural network for protein contact prediction, significantly improving the accuracy on blind, challenging targets with novel folds. RaptorX-Contact ranked first in free modeling targets at CASP12.[161] Its architecture (Figure 5(a)) entails (1) a 1D ResNet that inputs MSAs, predicted secondary structure and solvent accessibility (from DL-based prediction tool RaptorX-Property)[162] and (2) a 2D ResNet with dilations that inputs the 1D ResNet output and inter-residue co-evolution information from CCMpred.[144] In its original formulation, RaptorX-Contact outputs a binary classification of contacting versus non-contacting residue pairs.[102] Later versions were trained to learn multi-class classification for distance distributions between C_β atoms.[147] The primary contributors to the accuracy of predictions was the co-evolution information from CCMpred and the depth of the 2D ResNet, suggesting that the deep neural network learned co-evolution information better than previous methods. Later, the method was extended to predict $C_\alpha{-}C_\alpha$, $C_\alpha{-}C_\gamma$, $C_\gamma{-}C_\gamma$, N-O distances and torsion angles (DL-based RaptorX-Angle),[163] giving constraints to locate side chains and additionally constrain the backbone; all five distances, torsions, and secondary structure predictions were converted to constraints for folding by CNS.[147] At CASP12, however, RaptorX-Contact (original contact-based formulation) and DL drew limited attention because the difference between top-ranked predictions from DL-based methods and hybrid DCA-based methods was small.

This situation changed at CASP13 4 when one DL-based model, AlphaFold, developed by team A7D, or DeepMind,[26,22,164] ranked first and significantly improved the accuracy of "free modeling" (no templates available) targets (Figure 1). The A7D team modified the traditional simulated annealing protocol with DL-based predictions and tested three protocols based on deep neural networks. Two protocols used memory-augmented simulated annealing (with domain segmentation and fragment assembly) with potentials generated from predicted inter-residue distance distributions and predicted GDT_TS,[165] respectively, whereas the third protocol directly applies gradient descent optimization on a hybrid potential combining predicted distance and Rosetta score. For the distance prediction network, a deep ResNet, similar to that of RaptorX,[102] inputs MSA data and predicts the probability of distances between $\beta-$ carbons. A second network was

Table 3. A Summary of Structure Prediction Models

Model	Architecture	Dataset	N_train	Performance	Testset	Citation
/	MLP(2-layer)	proteases	13	3.0 Å RMSD (1TRM),1.2 Å RMSD (6PTI)	1TRM, 6PTI	Bohr et al.[9]
PSICOV	graphical Lasso	–	–	precision: Top-L 0.4, Top-L/2 0.53,Top-L/5 0.67, Top-L/10 0.73	150 Pfam	Jones et al.[141]
CMAPpro	2D biRNN + MLP	ASTRAL	2,352	precision: Top-L/5 0.31, Top-L/10 0.4	ASTRAL 1.75 CASP8, 9	Di Lena et al.[142]
DNCON	RBM	PDB SVMcon	1,230	precision: Top-L 0.46, Top-L/2 0.55, Top-L/5 0.65	SVMCON_TEST, D329, CASP9	Eickholt et al.[143]
CCMpred	LM	–	–	precision: Top-L 0.5, Top-L/2 0.6, Top-L/5 0.75, Top-L/10 0.8	150 Pfam	Seemayer et al.[144]
PconsC2	Stacked RF	PSICOV set	150	positive predictive value (PPV) 0.44	set of 383 CASP10(114)	Skwark et al.[145]
MetaPSICOV	MLP	PDB	624	precision: Top-L 0.54, Top-L/2 0.70, Top-L/5 0.83, Top-L/10 0.88	150 Pfam	Jones et al.[146]
RaptorX-Contact	ResNet	subset of PDB25	6,767	TM score: 0.518 (CCMpred: 0.333, MetaPSICOV: 0.377)	Pfam, CASP11, CAMEO, MP	Wang et al, 2017[102]
RaptorX-Distance	ResNet	subset of PDB25	6,767	TM score: 0.466 (CASP12), 0.551 (CAMEO), 0.474 (CASP13)	CASP12 + 13, CAMEO	Xu, 2018[147]
DeepCov	2D CNN	PDB	6,729	precision: Top-L 0.406, Top-L/2 0.523, Top-L/5 0.611, Top-L/10 0.642	CASP12	Jones et al, 2018[148]
SPOT	ResNet, Res-bi-LSTM	PDB	11,200	AUC: 0.958 (RaptorX-contact ranked 2nd: 0.909)	1,250 chains after June 2015	Hanson et al.[149]
DeepMetaPSICOV	ResNet	PDB	6,729	precision: Top-L/5 0.6618	CASP13	Kandathil et al, 2019[150]
MULTICOM	2D CNN	CASP 8-11	425	TM score: 0.69, GDT_TS: 63.54, SUM *Z* score (– 2.0): 99.47	CASP13	Hou et al.[151]
C-I-TASSER*	2D CNN	–	–	TM score: 0.67, GDT_HA: 0.44, RMSD: 6.19, SUM *Z* score(– 2.0): 107.59	CASP13	Zheng et al.[152]
AlphaFold	ResNet	PDB	31,247	TM score: 0.70, GDT_TS: 61.4,SUM *Z* score (– 2.0): 120.43	CASP13	Senior et al.[22]
MapPred	ResNet	PISCES	7,277	precision: 78.94% in SPOT, 77.06% in CAMEO, 77.05 in CASP12	SPOT, CAMEO, CASP12	Wu et al, 2019[153]
trRosetta	ResNet	PDB	15,051	TM_score: 0.625 (AlphaFold: 0.587)	CASP13, CAMEO	Yang et al, 2020[103]
RGN	bi-LSTM	ProteinNet 12 (before 2016)**	104,059	10.7 Å dRMSD on FM, 6.9 Å on TBM	CASP12	AlQuraishi, 2019[101]
/	biGRU, Res LSTM	CUProtein	75,000	preceded CASP12 winning team, comparable with AlphaFold in RMSD	CASP12 + 13	Drori et al.[78]

FM, free modeling; GRU, gated recurrent unit; LM, pseudo-likelihood maximization; MLP, multi-layer perceptron; MP, membrane protein; RBM, restricted Boltzmann machine; RF, random forest; RMSD, root-mean square deviation; TBM, template-based modeling.

*C-I-TASSER and C-QUARK were reported, we only report one here.

**RGN was trained on different ProteinNet for each CASP, we report the latest one here.

trained to predict GDT_TS of the candidate structure with respect to the true or native structure. The simulated annealing process was improved with a conditional variational autoencoder (CVAE)[166] model that constructs a mapping between the backbone torsions and a latent space conditioned by sequence. With this network, the team generated a database of nine-residue fragments for the memory-augmented simulated annealing system. Gradient-based optimization performed slightly better than the simulated annealing, suggesting that traditional simulated annealing is no longer necessary and state-of-the-art performance can be reached with simply optimizing a network predicted potential. AlphaFold's authors,

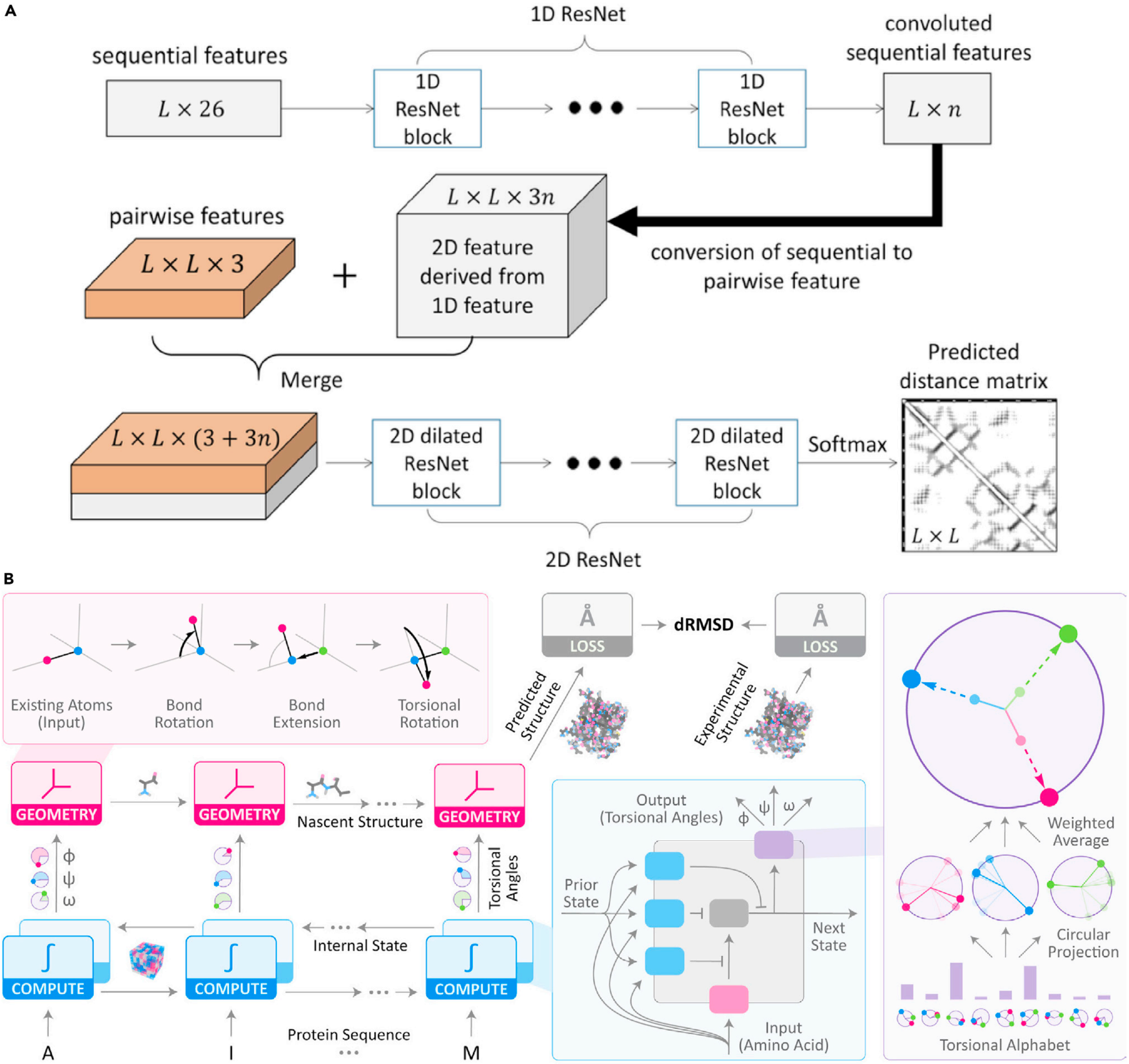

Figure 5. Two Representative DL Approaches to Protein Structure Prediction

(A) Residue distance prediction by RaptorX: the overall network architecture of the deep dilated ResNet used in CASP13. Inputs of the first-stage, 1D convolutional layers are a sequence profile, predicted secondary structure, and solvent accessibility. The output of the first stage is then converted into a 2D matrix by concatenation and fed into a deep ResNet along with pairwise features (co-evolution information, pairwise contact, and distance potential). A discretized inter-residue distance is the output. Additional network layers can be attached to predict torsion angles and secondary structures. Figure from Xu and Wang (2019).[160]

(B) Direct structure prediction: overview of recurrent geometric networks (RGN) approach. The raw amino acid sequence along with a PSSM are fed as input features, one residue at a time, to a bidirectional LSTM net. Three torsion angles for each residue are predicted to directly construct the 3D structure. Figure from AlQuraishi (2019).[101]

like the RaptorX-Contact group, emphasized that the accuracy of predictions relied heavily on learned distance distributions and co-evolutionary data.

Yang et al.[103] further improved the accuracy of predictions on CASP13 targets using a shallower network than former models (61 versus 220 ResNet blocks in AlphaFold) by also training their neural network model (named trRosetta) to learn inter-residue orientations along with $\beta-$ carbon distances. The geometric features—C_α-C_β torsions, pseudo-bond angles, and azimuthal rotations—directly describe the relevant coordinates for the physical interaction of two amino acid side chains. These additional outputs created significant improvement on a relatively fixed DL framework, suggesting that there is room for additional improvement.

An alternative and intuitive approach to structure prediction is directly learning the mapping from sequence to structure with a neural network. AlQuraishi[101] developed such an end-to-end differentiable protein structure predictor, called RGN, that allows direct prediction of torsion angles to construct the protein backbone (Figure 5B). RGN is a bidirectional LSTM that inputs a sequence, PSSM, and positional information and outputs predicted backbone torsions. Overall 3D structure predictions are within 1–2 Å of those made by top-ranked groups at CASP13, and this approach boasts a considerable advantage in prediction time compared with strategies that learn potentials. Moreover, the method does not use MSA-based information and could potentially be improved with the inclusion of evolutionary information. The RGN strategy is generalizable and well suited for protein structure prediction. Several generative methods (see below) also entail end-to-end structure prediction models, such as the CVAE framework used by AlphaFold, albeit with more limited success.[22]

Related Applications

Side-chain prediction is required for homology modeling and various protein engineering tasks, such as fixed-backbone design. Side-chain prediction is often embedded in high-resolution structure prediction methods, traditionally with dead-end elimination[167] or preferential sampling from backbone-dependent side-chain rotamer libraries.[168] Liu et al.[169] specifically trained a 3D CNN to evaluate the probability score for different potential rotamers. Du et al.[170] adopted an energy-based model (EBM)[171] to recover rotamers for backbone structures. Recent protein structure prediction models, such as Gao et al.'s[163] RaptorX-angle and Yang et al.'s[103] trRosetta, predict the structural features that help locate the position of side-chain atoms as well.

PPI prediction identifies residues at the interface of the two proteins forming a complex. Once the interface residues are determined, a local search and scoring protocol can be used to determine the structure of a complex. Similar to protein folding, efforts have focused on learning to classify contact or not. For example, Townshend et al.[96] developed a 3D CNN model (SASNet) that voxelizes the 3D environment around the target residue, and Fout et al.[112] developed a GCN-based model with each interacting partner represented as a graph. Unlike those starting from the unbound structures, Zeng et al.[172] reuse the model trained on single-chain proteins (i.e., RaptorX-Contact) to predict PPI with sequence information alone, which resulted in the RaptorX-Complex that outperforms ECA-based methods at contact prediction. Another interesting approach directly compares the geometry of two protein patches. Gainza et al.[116] trained their MaSIF model by minimizing the Euclidean distances between the complementary surface patches on the two proteins while maximizing the distances between non-interacting surface patches. This step is followed by a quick nearest-neighbor scanning to predict binding partners. The accuracy of MaSIF was comparable with traditional docking methods. However, MaSIF, similar to existing methods, showed low prediction accuracy for targets that involve conformational changes during binding.

Membrane proteins (MPs) are partially or fully embedded in a hydrophobic environment composed of a lipid bilayer and, consequently, they exhibit hydrophobic motifs on the surface unlike the majority of the proteins that are water soluble. Wang et al.[173] used a DL transfer learning framework comprising one-shot learning from non-MPs to MPs. They showed that transfer learning works surprisingly well here because the most frequently occurring contact patterns in soluble proteins and MPs are similar. Other efforts include classification of the trans-membrane topology.[174] Since experimental biophysical data are sparse for MPs, Alford and Gray[175] compiled a collection of 12 diverse benchmark sets for membrane protein prediction and design for testing and learning of implicit membrane energy models.

Loop modeling is a special case of structure prediction, where most of the 3D protein structure is given, but coordinates of segments of the polypeptide are missing and need to be completed. Loops are irregular and sometimes flexible segments, and thus their structures have been difficult to capture experimentally or computationally.[176,177] So far, DL frameworks based on inter-residue distance prediction (similar to protein structure prediction),[178] and those based on treating the loop residue distances with the remaining residues as an image inpainting problem[179] have been applied to loop modeling. Recently, Ruffolo et al.[177] used a RaptorX-like network setup and a trRosetta geometric representation to predict the structure of antibody hypervariable complementarity-determining region (CDR) H3 loops, which is critical for antigen binding.

PROTEIN DESIGN

We divide the current DL approaches to protein design into two broad categories. The first uses knowledge of other sequences (either "all" sequenced proteins or a certain class of proteins) to design sequences directly (Table 4). These approaches are well suited to create new proteins with functionality matching existing proteins based on sequence information alone, in a manner similar to consensus design.[180] The second class follows the "fold-before-function" scheme and seeks to stabilize specific 3D structures, perhaps but not necessarily with the intent to perform a desired function (Tables 5 and 6). The first approach can be described as function → sequence (structure agnostic), and the second approach fits the traditional stepwise inverse design: function → structure → sequence.

Many of the recent studies describe novel algorithms that output putative designed protein sequences, but only a few studies also present experimental validation. In traditional protein design studies, it is not uncommon for most designs to fail, and some of the early reports of protein designs were later withdrawn when the experimental evidence was not confirmed by others. As a result, it is usually expected that design studies offer rigorous experimental evidence. In this review, because we are interested in creative, emerging DL methods for design, we include papers that lack experimental validation, and many of these have *in silico* tests that help gauge validity. In addition, we make a special note of recent studies that present experimental validation of designs.

Direct Design of Sequence

Approaches that attempt to design for sequences parallel work in the field of NLP, where an auto-regressive framework is common, most notably, the RNN. In language processing, an RNN

Table 4. Generative Models to Identify Sequence from Function (Design for Function)

Model	Architecture	Output	Dataset	N_train	Performance	Citation
–	WGAN + AM	DNA	chromosome 1 of human hg 38	4.6M	~4 times stronger than training data in predicted TF binding	Killoran et al.[181]
–	VAE	AA	5 protein families	–	natural mutation probability prediction rho = 0.58	Sinai et al.[93]
–	LSTM	AA	ADAM, APD, DADP	1,554	predicted antimicrobial property 0.79 ± 0.25 (random: 0.63 ± 0.26)	Müller et al, 2018[55]
PepCVAE	CVAE	AA	–	15K labeled, 1.7M unlabeled	generate predicted AMP with 83% (random, 28%; length, 30)	Das et al.[64]
FBGAN	WGAN	DNA	UniProt (res., 50)	3,655	predicted antimicrobial property over 0.9 after 60 epochs	Gupta et al.[182]
DeepSequence	VAE	AA	mutational scan data	41 scans	aimed for mutation effect prediction, outperformed previous models	Riesselman et al.[94]
DbAS-VAE	VAE+AS	DNA	simulated data	–	predicted protein expression surpassed FB-GAN/VAE	Brookes et al.[183]
–	LSTM	musical scores	–	56 betas + 38 alphas	generated proteins capture the secondary structure feature	Yu et al.[184]
BioSeqVAE	VAE	AA	UniProt	200,000	83.7% reconstruction accuracy,70.6% EC accuracy	Costello et al.[185]
–	WGAN	AA	antibiotic resistance determinants	6,023	29% similar to training sequence (BLASTp)	Chhibbar et al.[186]
PEVAE	VAE	AA	3 protein families	31,062	latent space captures phylogenetic, ancestral relationship, and stability	Ding et al.[92]
–	ResNet	AA	mutation data + llama immune repertoire	1.2M (nano)	predicted mutation effect reached state-of-the-art, built a library of CDR3 seq	Riesselman et al.[187]
Vampire	VAE	AA	immuneACCESS	–	generated sequences predicted to be similar to real CDR3 sequences	Davidson et al, 2019[188]
ProGAN	CGAN	AA	eSol	2,833	solubility prediction R^2 improved from 0.41 to 0.45	Han et al, 2019[189]
ProteinGAN	GAN	AA	MDH from UniProt	16,706	60 sequences were tested *in vitro*, 19 soluble, 13 with catalytic activity	Repecka et al.[190]
CbAS-VAE	VAE+AS	AA	protein fluorescence dataset	5,000	predicted protein fluorescence surpassed FB-VAE/DbAS	Brookes et al.[183]

AA, amino acid sequence; AM, activation maximization; AS, adaptive sampling; CGAN, conditional generative adversarial network; CVAE, conditional variational autoencoder; DNA, DNA sequence; EC, enzyme commission.

model is able to take the beginning of a sentence and predict the next word in that sentence. Likewise, given a starting amino acid residue or a sequence of residues, a protein design model can output a categorical distribution for each of the 20 amino acid residues for the next position in the sequence. The next residue in the sequence is sampled from this categorical distribution, which in turn is used as the input to predict the following one. Following this approach, new sequences, sampled from the distribution of the training data, are generated, with the goal of having properties similar to those in the training set. Müller et al.[55] first applied an LSTM RNN framework to learn sequence patterns of antimicrobial peptides (AMPs),[204] a highly specialized sequence space of cationic, amphipathic helices. The same group then applied this framework to design membranolytic anticancer peptides.[205] Twelve of the generated peptides were synthesized and six of them killed MCF7 human breast adenocarcinoma cells with at least 3-fold selectivity against human erythrocytes. In another application, instead of traditional RNNs, Riesselman et al.[187] used a residual causal dilated CNN[206] in an auto-regressive way and generated a functional single-domain antibody library conditioned on the naive immune repertoires from llamas; although experimental validation was not presented. Such applications could potentially speed up and simplify the task of generating sequence libraries in the lab.

Another approach to sequence generation is mapping the latent space to the sequence space, and common strategies to train such a mapping include AEs and GANs. As mentioned earlier, AEs are trained to learn a bidirectional mapping between a discrete design space (sequence) and a continuous real-valued space (latent space). Thus, many applications of AEs use the learnt latent representation to capture the sequence distribution of a specific class of proteins, and subsequently, to predict the effect of variations in sequence (or mutations) on protein function.[92–94] The utility of this learned latent space, however, is more than that. A well trained real-valued latent space can be used to interpolate between two training samples, or even

Table 5. Generative Models for Protein Structure Design

Model	Architecture	Representation	Dataset	N_train	Performance	Citation
–	DCGAN	C_α-C_α distances	PDB (16-, 64-, 128-residue fragments)	115,850	meaningful secondary structure, reasonable Ramachandran plot	Anand et al.[24]
RamaNet	GAN	torsion angles	ideal helical structures from PDB	607	generated torsions are concentrated around helical region	Sabban et al.[191]
–	DCGAN	backbone distance	PDB (64-residue fragment)	800,000	smooth interpolations; recover from sequence design and folding	Anand et al.[68]
Ig-VAE	VAE	coordinates and backbone distance	AbDb (antibody structure)	10,768	sampled 5,000 Igs screened for SARS-CoV2 Binder	Eguchi et al.[192]
–	CNN (input design)	same as trRosetta	–	–	27 out of 129 sequence-structure pairs experimentally validated	Anishchenko et al.[193]

CNN, convolutional neural network; DCGAN, deep convolutional generative adversarial network; GAN, generative adversarial network; VAE, variational autoencoder.

extrapolate beyond the training data to yield novel sequences. One such example is the PepCVAE model.[64] Following a semi-supervised learning approach, Das et al.[64] trained a VAE model on an unlabeled dataset of 1.7×10^6 sequences and then refined the model for the AMP subspace using a 15,000 sequence-labeled dataset. By concatenating a conditional code indicating if a peptide is antimicrobial, the CVAE framework allows efficient sampling of AMPs selectively from the broader peptide space. More than 82% of the generated peptides were predicted to exhibit antimicrobial properties according to a state-of-the-art AMP classifier.

Unlike AEs, GANs focus on learning the unidirectional mapping from a continuous real-valued space to the design space. In an early example, Killoran et al.'s[181] developed a model that combines a standard GAN and activation maximization to design DNA sequences that bind to a specific protein. Repecka et al.[190] trained ProteinGAN on the bacterial enzyme malate dehydrogenase (MDH) to generate new enzyme sequences that were active and soluble *in vitro*, some with over 100 mutations, with a 24% success rate. Another interesting GAN-based framework is Gupta and Zou's[207] FeedBack GAN (FBGAN) that learns to generate cDNA sequences for peptides. They add a feedback-loop architecture to optimize the synthetic gene sequences for desired properties using an oracle (an external function analyzer). At every epoch, they update the positive training data for the discriminator with high-scoring sequences from the generator so that the score of generated sequences increases gradually. They demonstrated the efficacy of their model by successfully biasing generated sequences toward antimicrobial activity and a desired secondary structure.

Design with Structure as Intermediate

Within the fold-before-function scheme, for design one first picks a protein fold or topology according to certain desirable properties, then determines the amino acid sequence that could fold into that structure (function → structure →sequence). Under the supervised learning setting, most efforts use the native sequences as the ground truth and recovery rate of native sequences (i.e., the percentage of sequence that matches the native one) as a success metric. To compare, Kuhlman and Baker[208] reported sequence recovery rates of 51% for core residues and 27% for all amino acid residues using traditional *de novo* design approaches. Because the mapping from sequence to structure is not unique (within a neighborhood of each structure), it is not clear that higher sequence recovery rates would be meaningful.

A class of efforts, pioneered by the SPIN model,[209] inputs a five-residue sliding window to predict the amino acid probabilities for the center position to generate sequences compatible with a desired structure. The features in such models include φ and ψ dihedrals, a sequence profile of a five-residue fragment derived from similar structures, and a rotamer-based energy profile of the target residue using the DFIRE potential. SPIN[209] reached a 30.7% sequence recovery rate and Wang et al.[194] and O'Connell et al.'s[25] SPIN2 further improved it to 34%. Another class of efforts inputs the voxelized local environment of an amino acid residue. In Zhang et al.'s[197,198] and Shroff et al.'s[197,198] models, voxelized local environment was fed into a 3D CNN framework to predict the most stable residue type at the center of a region. Shroff et al.[198] reported a 70% recovery rate and the mutation sites were validated experimentally. Anand et al.[202] trained a similar model to design sequences for a given backbone. Their protocol involves iteratively sampling from predicted conditional distributions, and it recovered from 33% to 87% of native sequence identities. They tested their model by designing sequences for five proteins, including a *de novo* TIM barrel. The designed sequences were 30%–40% identical to native sequences and predicted structures were 2–5 Å root-mean-square deviation from the native conformation.

Other approaches generate full sequences conditioned by a target structure. Greener et al.[195] trained a CVAE model to generate sequences conditioned on protein topology represented in a string.[99] The resulting sequence was verified to be stable with molecular simulation. Karimi et al.[210] developed gcWGAN that combined a CGAN and a guidance strategy to bias the generated sequences toward a desired structure. They used a fast structure prediction algorithm[211] as an "oracle"

Table 6. Generative Models to Identify Sequence from Structure (Protein Design)

Model	Architecture	Input	Dataset	N_train	Performance	Citation
SPIN	MLP	sliding window with 136 features	PISCES	1,532	sequence recovery of 30.7% on 1,532 proteins (CV)	Li et al.[100]
SPIN2	MLP	sliding window with 190 features	PISCES	1,532	sequence recovery of 34.4% on 1,532 proteins (CV)	O'Connell et al.[25]
–	MLP	target residue and its neighbor as pairs	PDB	10,173	sequence recovery of 34% on 10,173 proteins	Wang et al.[194]
–	CVAE	string encoded structure or metal	PDB, MetalPDB	3,785	verified with structure prediction and dynamic simulation	Greener et al.[195]
SPROF	Bi-LSTM + 2D ResNet	112 1-D features + C_α distance map	PDB	11,200	sequence recovery of 39.8% on protein	Chen et al.[196]
ProDCoNN	3D CNN	gridded atomic coordinates	PDB	17,044	sequence recovery of 42.2% on 5,041 proteins	Zhang et al.[197]
–	3D CNN	gridded atomic coordinates	PDB-REDO	19,436	sequence recovery 70%, experimental validation of mutation	Shroff et al.[198]
ProteinSolver	Graph NN	partial sequence, adjacency matrix	UniParc	72×10^6 residues	sequence recovery of 35%, folding and MD test with 4 proteins	Strokach et al, 2019[199]
gcWGAN	CGAN	random noise + structure	SCOPe	20,125	diversity and TM score of prediction from designed sequence $\geq$cVAE	Karimi et al.[200]
–	Graph Transformer	backbone structure in graph	CATH based	18,025	perplexity: 6.56 (rigid), 11.13 (flexible) (random: 20.00)	Ingraham et al.[23]
DenseCPD	ResNet	gridded backbone atomic density	PISCES	2.6×10^6 residues	sequence recovery of 54.45% on 500 proteins	Qi et al.[201]
–	3D CNN	gridded atomic coordinates	PDB	21,147	sequence recovery from 33% to 87%, test with folding of TIM barrel	Anand et al.[202]
–	CNN (input design)	Same as trRosetta	–	–		Norn et al.[203]

Bi-LSTM, bidirectional long short-term memory; CV, cross-validation; MLP, multi-layer perceptron.

to assess the output sequence and provide feedback to refine the model. They examined the model for six folds using Rosetta-based structure prediction, and gcWGAN had higher TM score distributions and more diverse sequence profiles than CVAE.[195] Another notable experiment is Ingraham et al.'s[23] graph transformer model that inputs a structure, represented as a graph, and outputs the sequence profile. They treat the sequence design problem similar to a machine translation problem, i.e., a translation from structure to sequence. Like the original transformer model,[57] they adopted an encoder-decoder framework with self-attention mechanisms to dynamically learn the relationship between information in two neighbor layers. They measured their results by perplexity, a widely used metric in speech recognition,[212] and the per-residue perplexity (lower is better) for single chains was 9.15, lower than the perplexity for SPIN2 (12.86). Norn et al. treated the protein design problem as that of maximizing the probability of a sequence given a structure. They back-propagate through the trRosetta structure prediction network[103] to find a sequence that minimizes the distance between predicted structure and a desired structure.[203] Norn et al. validate their designs computationally by showing the generated sequences have deep wells in their modeled energy landscapes. Strokach et al. treated the design of protein sequence given a target structure as a constraint satisfaction problem. They optimized their GNN architecture on the related problem of filling in a Sudoku puzzle followed by training on millions of protein sequences corresponding to thousands of structural folds. They were able to validate designed sequences *in silico* and demonstrate that some designs folded to their target structures *in vitro*.[213]

An ambitious design goal is to generate new structures without specifying the target structure. Anand and Huang were the first to generate new structures using DL. They tested various representations (e.g., full atom, torsion-only) with a deep convolutional GAN (DCGAN) framework that generates sequence-agnostic, fixed-length short protein structural fragments.[24] They found that the distance map of C_α atoms gives the most meaningful protein structures, although the asymmetry of ψ and φ torsion angles[214] was only recovered with torsion-based representations. Later they extended this work to all atoms in the backbone and combined with a recovery network to avoid the time-consuming structure reconstruction process.[68] They showed that some of the designed folds are stable in molecular simulation. In a more narrowly focused study, Eguchi et al.[192] trained a VAE model with the structures of immunoglobulin (Ig) proteins, called Ig-VAE. By sampling the latent space, they generated 5,000 new Ig structures (sequence-agnostic) and then screened them with computational docking to identify putative binders to SARS-CoV2-RBD.

Another approach exploits a DL structure prediction algorithm and a Markov Chain MC (MCMC) search to find sequences that fold into novel compact structures. Anishchenko et al.[193] iterated sequences through the DL network, trRosetta,[103] to "hallucinate"[215] mutually compatible sequence-structure pairs in a manner similar to "input design".[183] By maximizing the contrast between the distance distributions predicted by trRosetta and a background network trained on noise, they obtained new sequences with geometric maps with sharp geometric features. Impressively, 27 of the[127] hallucinated sequences were experimentally validated to fold into monomeric, highly stable, proteins with circular dichroism spectra compatible with the predicted structure.

OUTLOOK AND CONCLUSION

In this review, we have summarized the current state-of-the-art DL techniques applied to the problem of protein structure prediction and design. As in many other areas, DL shows the potential to revolutionize the field of protein modeling. While DL originated from computer vision, NLP and machine learning, its fast development combined with knowledge from operations research,[216] game theory,[65] and variational inference[32] among other fields, has resulted in many new and powerful frameworks to solve increasingly complex problems. The application of DL for biomolecular structure has just begun, and we expect to see more efforts on methodology development and applications in protein modeling and design.

We observed several trends.

Experimental Validation

An important gap in current DL work in protein modeling, especially protein design (with few notable exceptions),[205,190,198,193] is the lack of experimental validation. Past blind challenges, e.g., CASP and CAPRI, and design claims have shown that experimental validation in this field is of paramount importance, where computational models are still prone to error. A key next stage for this field is to engage collaborations between machine learning experts and experimental protein engineers to test and validate these emerging approaches.

Importance of Benchmarking

In other fields of machine learning, standardized benchmarks have triggered rapid progress.[217–219] CASP is a great example that provides a standardized platform for benchmarking diverse algorithms, including emerging DL-based approaches. A well-defined question and proper evaluation (especially experimental) would lead to more open competition among a broader range of groups and, eventually, the innovation of more diverse and powerful algorithms.

Imposing a Physics-Based Prior

One common topic among the machine learning community is how to utilize existing domain knowledge to reduce the effort during training. Unlike certain classical ML problems, such as image classification, in protein modeling, a wide range of biophysical principles restrict the range of plausible solutions. Some examples in related fields include imposing a physics-based model prior,[220,221] adding a regularization term with physical meaning,[222] and adopting a specific formula to conserve physical symmetry.[223,224] Similarly, in protein modeling, well-established empirical observations can help restrict the solution space, such as the Ramanchandran distribution of backbone torsion angles[214] and the Dunbrack or Richardsons library of side-chain conformations.[225,226]

Closed-Loop Design

The performance of DL methodologies relies heavily on the quality of data, but the publicly available datasets may not cover important sample space because of experimental accessibility at the time of experiments. Furthermore, the dataset may contain harmful noise from non-uniform experimental protocols and conditions. A possible solution may be to combine model training with experimental data generation. For instance, one may devise a closed-loop strategy to generate experimental data, on-the-fly, for queries (or model inputs) that are most likely to improve the model, and update the training dataset with the newly generated data.[227–230] For such a strategy to be feasible, automated synthesis and characterization is necessary. As high-throughput synthesis and testing of protein (or DNA and RNA) can be carried out in parallel, automation is possible. While such a strategy may seem far-fetched, automated platforms such as those from Ginkgo Bioworks or Transcriptic are already on the market.

Reinforcement Learning

Another approach to overcome the limitation of data availability is reinforcement learning (RL). Biologically meaningful data may be generated on-the-fly in simulated environments, such as the Foldit game. In the most famous application of RL, AlphaGo Zero,[21] an RL agent (network) was able to learn and master the game by learning from the game environment alone. There are already some examples of RL in the field of chemistry and electric engineering to optimize organic molecules or computational chips.[231–233] One suitable protein modeling problem for an RL algorithm would be training an artificial intelligence (AI) agent to make a series of "moves" to fold a protein, similar to the Foldit game.[234,235] Such studies are still rare and previous attempts have focused on folding the 2D hydrophobic-polar model of proteins.[236,237] Although the results did not yet beat conventional methods, Gao[238] recently explored using policy and reward networks in an RL scheme to fold 3D protein structures *de novo* by guiding the selection of MC moves in Rosetta. Angermueller et al.[239] applied a model-based RL framework to designing sequences of AMPs and transcription factor binding sites.

Model Interpretability

One should keep in mind that a neural network represents nothing more (and nothing less) than a powerful and flexible regression model. In addition, due to their highly recursive nature, neural networks tend to be regarded as "black-boxes", i.e., too complicated for practitioners to understand the resulting parameters and functions. Although model interpretability in ML is a rapidly developing field, many popular approaches, such as saliency analysis[240–242] for image classification models, are far from satisfactory.[243] Although other approaches[244,245] offer more reliable interpretations, their application to DL model interpretation has been largely missing in protein modeling. As a result, current DL models offer limited understanding of the complex patterns they learn.

Beyond Proteins

DL-based methods are general and so, with appropriate representation and sufficient training data, they can be applied to other molecules. Like proteins, nucleic acids, carbohydrates, and lipids are also polymers, composed of nucleotides, monosaccharides, and aliphatic subunits and head groups, respectively. Many approaches developed for learning protein sequence and structural information can be extended to these other classes of biomolecules.[246,247] Finally, biology often conjugates these molecules, e.g., for glycoproteins. DL approaches that build up from basic chemistry, such as those being developed for small molecule drugs,[248–251] may inspire methods to treat these biomolecules that do not fall into a strict polymer type.

The "Sequence → Structure → Function" Paradigm

We know from molecular biophysics that a sequence translates into function through the physical intermediary of a 3D molecular structure. Allosteric proteins,[252] for instance, may exhibit different structural conformations under different physiological conditions (e.g., pH) or environmental stimuli (e.g., small molecules, inhibitors), reminding us that context is as important as protein sequence. That is, despite Anfinsen's[42] hypothesis, sequence alone does not always fully determine the structure. Some proteins require chaperones to fold to their native structure, meaning that a sequence could result in non-native conformations when the kinetics of folding to the native structure may be unfavorable in the absence of a chaperone. Because many powerful DL algorithms in NLP operate on sequential data, it may seem reasonable to use protein sequences alone for training DL models. In principle, with a suitable framework and training, DL could disentangle the underlying relationships between sequence and structural elements. However, a careful selection of DL frameworks that are structure or mechanism-aware will accelerate learning and improve predictive power. Indeed, many successful DL frameworks applied so far (e.g., CNNs or graph CNNs) factor in the importance of learning on structural information.

Finally, with the hope of gaining insight into the fundamental science of biomolecules, there is a desire to link AI approaches to the underlying biochemical and biophysical principles that drive biomolecular function. For more practical purposes, a deeper understanding of underlying principles and hidden patterns that lead to pathology is important in the development of therapeutics. Thus, while efforts strictly limited to sequences are abundant, we believe that models with structural insights will play a more critical role in the future.

ACKNOWLEDGMENTS

This work was supported by the NIH through grant R01-GM078221. We thank Dr. Justin S. Smith at the Center for Nonlinear Studies at Los Alamos National Laboratory, NM, for helpful discussion and Dr. Andrew D. White at the Department of Chemical Engineering at University of Rochester, NY, and Alexander Rives at the Department of Computer Science at New York University, NY, for helpful suggestions. We are also grateful for insightful suggestions from the reviewers.

AUTHOR CONTRIBUTIONS

Conceptualization, W.G. and J.J.G.; Investigation, W.G. and S.P.M.; Writing – Original Draft, W.G.; Writing – Review & Editing, W.G., S.P.M., J.S., and J.J.G.; Funding Acquisition, J.J.G.; Resources, J.J.G.; Supervision, J.S. and J.J.G.

REFERENCES

1. Slabinski, L., Jaroszewski, L., Rodrigues, A.P., Rychlewski, L., Wilson, I.A., Lesley, S.A., and Godzik, A. (2007). The challenge of protein structure determination-lessons from structural genomics. Protein Sci. *16*, 2472–2482.

2. Markwick, P.R.L., Malliavin, T., and Nilges, M. (2008). Structural biology by NMR: structure, dynamics, and interactions. PLoS Comput. Biol. *4*, e1000168.

3. Jonic, S., and Vénien-Bryan, C. (2009). Protein structure determination by electron cryo-microscopy. Curr. Opin. Pharmacol. *9*, 636–642.

4. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., and Moult, J. (2019). Critical assessment of methods of protein structure prediction (CASP)—Round XIII. Proteins *87*, 1011–1020.

5. Hollingsworth, S.A., and Dror, R.O. (2018). Molecular dynamics simulation for all. Neuron *99*, 1129–1143.

6. Ranjan, A., Fahad, M.S., Fernandez-Baca, D., Deepak, A., and Tripathi, S. (2019). Deep robust framework for protein function prediction using variable-length protein sequences. IEEE/ACM Trans. Comput. Biol. Bioinform. *17*, 1648–1659.

7. Huang, P.S., Boyken, S.E., and Baker, D. (2016). The coming of age of de novo protein design. Nature *537*, 320–327.

8. Yang, K.K., Wu, Z., and Arnold, F.H. (2019). Machine-learning-guided directed evolution for protein engineering. Nat. Methods *16*, 687–694.

9. Bohr, H., Bohr, J., Brunak, S., Cotterill, J.R., Fredholm, H., Lautrup, B., and Petersen, S. (1990). A novel approach to prediction of the 3-dimensional structures of protein backbones by neural networks. FEBS Lett. *261*, 43–46.

10. Schneider, G., and Wrede, P. (1994). The rational design of amino acid sequences by artificial neural networks and simulated molecular evolution: de novo design of an idealized leader peptidase cleavage site. Biophys. J. *66*, 335–344.

11. Schneider, G., Schrödl, W., Wallukat, G., Müller, J., Nissen, E., Rönspeck, W., Wrede, P., and Kunze, R. (1998). Peptide design by artificial neural networks and computer-based evolutionary search. Proc. Natl. Acad. Sci. U S A *95*, 12179–12184.

12. Ofran, Y., and Rost, B. (2003). Predicted protein-protein interaction sites from local sequence information. FEBS Lett. *544*, 236–239.

13. Nielsen, M., Lundegaard, C., Worning, P., Lauemøller, S.L., Lamberth, K., Buus, S., Brunak, S., and Lund, O. (2003). Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci. *12*, 1007–1017.

14. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature *521*, 436–444.

15. Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. Mol. Syst. Biol. *12*, 878.

16. Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., Kalinin, A.A., Do, B.T., Way, G.P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M.M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. J. R. Soc. Interfaces *15*, 20170387.

17. Mura, C., Draizen, E.J., and Bourne, P.E. (2018). Structural biology meets data science: does anything change? Curr. Opin. Struct. Biol. *52*, 95–102.

18. Noé, F., De Fabritiis, G., and Clementi, C. (2020). Machine learning for protein folding and dynamics. Curr. Opin. Struct. Biol. *60*, 77–84.

19. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M.S. (2016). Deep learning for visual understanding: a review. Neurocomputing *187*, 27–48.

20. Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. IEEE Comput. Intelligence Mag. *13*, 55–75.

21. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. Nature *1550*, 354.

22. Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Chongli, Q., Žídek, A., Nelson, A.W.R., Bridgland, A., et al. (2019). Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). Proteins *87*, 1141–1148.

23. Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. (2019). Generative models for graph-based protein design. Adv. Neural Inf. Process. Syst. 15820–15831.

24. Anand, N., and Huang, P. (2018). Generative modeling for protein structures. Adv. Neural Inf. Process. Syst. 7494–7505.

25. O'Connell, J., Li, Z., Hanson, J., Heffernan, R., Lyons, J., Paliwal, K., Dehzangi, A., Yang, Y., and Zhou, Y. (2018). SPIN2: predicting sequence profiles from protein structures using deep neural networks. Proteins: Struct. Funct. Bioinformatics *86*, 629–633.

26. Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A.W., Bridgland, A., et al. (2020). Improved protein structure prediction using potentials from deep learning. Nature, 1–5.

27. Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y., and Gao, X. (2019). Deep learning in bioinformatics: introduction, application, and perspective in the big data era. Methods *166*, 4–21.

28. Noé, F., Tkatchenko, A., Müller, K.-R., and Clementi, C. (2020). Machine learning for molecular simulation. Annu. Rev. Phys. Chem. *71*, 361–390.

29. Graves, J., Byerly, J., Priego, E., Makkapati, N., Parish, S.V., Medellin, B., and Berrondo, M. (2020). A review of deep learning methods for antibodies. Antibodies *9*, 12.

30. Kandathil, S.M., Greener, J.G., and Jones, D.T. (2019). Recent developments in deep learning applied to protein structure prediction. Proteins: Struct. Funct. Bioinformatics *87*, 1179–1189.

31. Torrisi, M., Pollastri, G., and Le, Q. (2020). Deep learning methods in protein structure prediction. Comput. Struct. Biotechnol. J. *18*, 1301–1310.

32. Kingma, D.P., and Welling, M. (2013). Auto-encoding variational Bayes. arXiv *1312*, 6114.

33. Pauling, L., and Niemann, C. (1939). The structure of proteins. J. Am. Chem. Soc. *61*, 1860–1867.

34. Kuhlman, B., and Bradley, P. (2019). Advances in protein structure prediction and design. Nat. Rev. Mol. Cell Biol. *20*, 681–697.

35. UniProt-Consortium. (2019). UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res. *47*, D506–D515.

36. Kuhlman, B., Dantas, G., Ireton, G.C., Varani, G., Stoddard, B.L., and Baker, D. (2003). Design of a novel globular protein fold with atomic-level accuracy. Science *302*, 1364–1368.

37. Fisher, M.A., McKinley, K.L., Bradley, L.H., Viola, S.R., and Hecht, M.H. (2011). De novo designed proteins from a library of artificial sequences function in Escherichia coli and enable cell growth. PLoS One *6*, e15364.

38. Correia, B.E., Bates, J.T., Loomis, R.J., Baneyx, G., Carrico, C., Jardine, J.G., Rupert, P., Correnti, C., Kalyuzhniy, O., Vittal, V., et al. (2014). Proof of principle for epitope-focused vaccine design. Nature *507*, 201.

39. King, N.P., Sheffler, W., Sawaya, M.R., Vollmar, B.S., Sumida, J.P., André, I., Gonen, T., Yeates, T.O., and Baker, D. (2012). Computational design of self-assembling protein nanomaterials with atomic level accuracy. Science *336*, 1171–1174.

40. Tinberg, C.E., Khare, S.D., Dou, J., Doyle, L., Nelson, J.W., Schena, A., Jankowski, W., Kalodimos, C.G., Johnsson, K., Stoddard, B.L., et al. (2013). Computational design of ligand-binding proteins with high affinity and selectivity. Nature *501*, 212–216.

41. Joh, N.H., Wang, T., Bhate, M.P., Acharya, R., Wu, Y., Grabe, M., Hong, M., Grigoryan, G., and DeGrado, W.F. (2014). De novo design of a transmembrane Zn^{2+}-transporting four-helix bundle. Science *346*, 1520–1524.

42. Anfinsen, C.B. (1973). Principles that govern the folding of protein chains. Science *181*, 223–230.

43. Levinthal, C. (1968). Are there pathways for protein folding? J. Chim. Phys. *65*, 44–45.

44. Li, B., Fooksa, M., Heinze, S., and Meiler, J. (2018). Finding the needle in the haystack: towards solving the protein-folding problem computationally. Crit. Rev. Biochem. Mol. Biol. *53*, 1–28.

45. Dahiyat, B.I., and Mayo, S.L. (1997). De novo protein design: fully automated sequence selection. Science *278*, 82–87.

46. Korendovych, I.V., and DeGrado, W.F. (2020). De novo protein design, a retrospective. Q. Rev. Biophys. *53*, https://doi.org/10.1017/S0033583519000131.

47. Dougherty, M.J., and Arnold, F.H. (2009). Directed evolution: new parts and optimized function. Curr. Opin. Biotechnol. *20*, 486–491.

48. Sun, R. (2019). Optimization for deep learning: theory and algorithms. arXiv *1912*, 08957.

49. Schmidhuber, J. (2015). Deep learning in neural networks: an overview. Neural Networks *61*, 85–117.

50. LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., and Jackel, L.D. (1990). Handwritten digit recognition with a back-propagation network. Adv. Neural Inf. Process. Syst. 396–404.

51. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, 770–778.

52. Jordan, M.I. (1997). Serial order: a parallel distributed processing approach. Adv. Psychol. *121*, 471–495.

53. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Comput. *9*, 1735–1780.

54. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv *1406*, 1078.

55. Müller, A.T., Hiss, J.A., and Schneider, G. (2018). Recurrent neural network model for constructive peptide design. J. Chem. Inf. Model. *58*, 472–479.

56. Bahdanau, D.; Cho, K.H.; Bengio, Y. Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings. 2015.

57. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. Adv. Neural Inf. Process. Syst. *2017*, 5999–6009.

58. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: pretraining of deep bidirectional transformers for language understanding. arXiv *1810*, 04805.

59. Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C.L., et al. (2019). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. bioRxiv, 622803, https://doi.org/10.1101/622803. https://www.biorxiv.org/content/10.1101/622803v3.

60. Pittala, S., and Bailey-Kellogg, C. (2020). Learning context-aware structural representations to predict antigen and antibody binding interfaces. Bioinformatics *36*, 3996–4003.

61. Hinton, G.E., and Zemel, R.S. (1994). Autoencoders, minimum description length and Helmholtz free energy. Adv. Neural Inf. Process. Syst. 3–10.

62. Kingma, D.P., and Welling, M. (2019). An introduction to variational autoencoders. arXiv *1906*, 02691.

63. Blei, D.M., Kucukelbir, A., and McAuliffe, J.D. (2017). Variational inference: a review for statisticians. J. Am. Stat. Assoc. *112*, 859–877.

64. Das, P., Wadhawan, K., Chang, O., Sercu, T., Santos, C.D., Riemer, M., Chenthamarakshan, V., Padhi, I., and Mojsilovic, A. (2018). PepCVAE: semi-supervised targeted design of antimicrobial peptide sequences. arXiv *1810*, 07743.

65. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. Adv. Neural Inf. Process. Syst. 2672–2680. https://papers.nips.cc/paper/5423-generative-adversarial-nets.

66. Arjovsky, M., Chintala, S., Bottou, L., and Wasserstein, G.A.N. (2017). arXiv *1701*, 07875.

67. Kurach, K., Lučić, M., Zhai, X., Michalski, M., and Gelly, S. (2019). A large-scale study on regularization and normalization in GANs. Int. Conf. Mach. Learn. 3581–3590.

68. Anand, N., Eguchi, R., and Huang, P.-S. (2019). Fully differentiable full-atom protein backbone generation. Int. Conf. Learn. Rep. *35*. https://openreview.net/revisions?id=SJxnVL8YOV.

69. Niepert, M., Ahmed, M., and Kutzkov, K. (2016). Learning convolutional neural networks for graphs. Int. Conf. Mach. Learn. 2014–2023.

70. Luo, F., Wang, M., Liu, Y., Zhao, X.-M., and Li, A. (2019). DeepPhos: prediction of protein phosphorylation sites with deep learning. Bioinformatics *35*, 2766–2773.

71. Li, F., Chen, J., Leier, A., Marquez-Lago, T., Liu, Q., Wang, Y., Revote, J., Smith, A.I., Akutsu, T., Webb, G.I., et al. (2020). DeepCleave: a deep learning predictor for caspase and matrix metalloprotease substrates and cleavage sites. Bioinformatics *36*, 1057–1065.

72. Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intelligence *35*, 1798–1828.

73. Romero, P.A., Krause, A., and Arnold, F.H. (2013). Navigating the protein fitness landscape with Gaussian processes. Proc. Natl. Acad. Sci. U S A *110*, E193–E201.

74. Bedbrook, C.N., Yang, K.K., Rice, A.J., Gradinaru, V., and Arnold, F.H. (2017). Machine learning to design integral membrane channel rhodopsins for efficient eukaryotic expression and plasma membrane localization. PLoS Comput. Biol. *13*, e1005786.

75. Ofer, D., and Linial, M. (2015). ProFET: feature engineering captures high-level protein functions. Bioinformatics *31*, 3429–3436.

76. Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M. (2007). AAindex: amino acid index database, progress report 2008. Nucleic Acids Res. *36*, D202–D205.

77. Wang, S., Peng, J., Ma, J., and Xu, J. (2016). Protein secondary structure prediction using deep convolutional neural fields. Sci. Rep. *6*, 18962.

78. Drori, I., Thaker, D., Srivatsa, A., Jeong, D., Wang, Y., Nan, L., Wu, F., Leggas, D., Lei, J., Lu, W., et al. (2019). Accurate protein structure prediction by embeddings and deep learning representations. arXiv *1911*, 05531.

79. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv *1301*, 3781.

80. Le, Q., and Mikolov, T. (2014). Distributed representations of sentences and documents. Int. Conf. Mach. Learn. 1188–1196.

81. Asgari, E., and Mofrad, M.R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS One *10*, e0141287.

82. El-Gebali, S., Mistry, J., Bateman, A., Eddy, S.R., Luciani, A., Potter, S.C., Qureshi, M., Richardson, L.J., Salazar, G.A., Smart, A., et al. (2019). The Pfam protein families database in 2019. Nucleic Acids Res. *47*, D427–D432.

83. Cai, C., Han, L., Ji, Z.L., Chen, X., and Chen, Y.Z. (2003). SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res. *31*, 3692–3697.

84. Aragues, R., Sali, A., Bonet, J., Marti-Renom, M.A., and Oliva, B. (2007). Characterization of protein hubs by inferring interacting motifs from protein interactions. PLoS Comput. Biol. *3*, e178.

85. Yu, C., van der Schaar, M., and Sayed, A.H. (2016). Distributed learning for stochastic generalized Nash equilibrium problems. CoRR. https://doi.org/10.1109/TSP.2017.2695451.

86. Yang, K.K., Wu, Z., Bedbrook, C.N., and Arnold, F.H. (2018). Learned protein embeddings for machine learning. Bioinformatics *34*, 2642–2648.

87. Alley, E.C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G.M. (2019). Unified rational protein engineering with sequence-based deep representation learning. Nat. Methods *16*, 1315–1322.

88. Krause, B., Lu, L., Murray, I., and Renals, S. (2016). Multiplicative LSTM for sequence modelling. arXiv *1609*, 07959.

89. Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., and Rost, B. (2019). Modeling aspects of the language of life through transfer-learning protein sequences. BMC Bioinformatics *20*, 723.

90. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv *1802*, 05365.

91. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. arXiv *2005*, 14165.

92. Ding, X., Zou, Z., and Brooks, C.L., Iii (2019). Deciphering protein evolution and fitness landscapes with latent space models. Nat. Commun. *210*, 1–13.

93. Sinai, S., Kelsic, E., Church, G.M., and Nowak, M.A. (2017). Variational auto-encoding of protein sequences. arXiv *1712*, 03346.

94. Riesselman, A.J., Ingraham, J.B., and Marks, D.S. (2018). Deep generative models of genetic variation capture the effects of mutations. Nat. Methods *15*, 816–822.

95. Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., et al. (2019). Evaluating protein transfer learning with TAPE. Adv. Neural Inf. Process. Syst. 9689–9701. http://papers.nips.cc/paper/9163-evaluating-protein-transfer-learning-with-tape.

96. Townshend, R., Bedi, R., and Dror, R.O. (2018). Generalizable protein interface prediction with end-to-end learning. arXiv *1807*, 01297.

97. Simonovsky, M., and Meyers, J. (2020). DeeplyTough: learning structural comparison of protein binding sites. J. Chem. Inf. Model. *60*, 2356–2366.

98. Kolodny, R., Koehl, P., Guibas, L., and Levitt, M. (2002). Small libraries of protein fragments model native protein structures accurately. J. Mol. Biol. *323*, 297–307.

99. Taylor, W.R.A. (2002). "periodic table" for protein structures. Nature *416*, 657–660.

100. Li, J., and Koehl, P. (2014). 3D representations of amino acids–applications to protein sequence comparison and classification. Comput. Struct. Biotechnol. J. *11*, 47–58.

101. AlQuraishi, M. (2019). End-to-End differentiable learning of protein structure. Cell Syst. *8*, 292–301.e3.

102. Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2017). Accurate de novo prediction of protein contact map by ultra-deep learning model. PLoS Comput. Biol. *13*, e1005324.

103. Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., and Baker, D. (2020). Improved protein structure prediction using predicted interresidue orientations. Proc. Natl. Acad. Sci. U S A *117*, 1496–1503.

104. Brunger, A.T. (2007). Version 1.2 of the crystallography and NMR system. Nat. Protoc. *2*, 2728.

105. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., and Sun, M. (2018). Graph neural networks: a review of methods and applications. arXiv *1812*, 08434.

106. Ahmed, E., Saint, A., Shabayek, A., Cherenkova, K., Das, R., Gusev, G., Aouada, D., and Ottersten, B. (2018). Deep learning advances on different 3D data representations: a survey. arXiv *1*, 01462.

107. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S.Y. (2020). A comprehensive survey on graph neural networks. IEEE Trans. Neural Networks Learn. Syst. 1–21. https://ieeexplore.ieee.org/abstract/document/9046288.

108. Vishveshwara, S., Brinda, K., and Kannan, N. (2002). Protein structure: insights from graph theory. J. Theor. Comput. Chem. *1*, 187–211.

109. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., and Leskovec, J. (2018). Hierarchical graph representation learning with differentiable pooling. Adv. Neural Inf. Process. Syst. 4800–4810. https://papers.nips.cc/paper/7729-hierarchical-graph-representation-learning-with-differentiable-pooling.

110. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S., Smola, A.J., and Kriegel, H.-P. (2005). Protein function prediction via graph kernels. Bioinformatics *21*, i47–i56.

111. Dobson, P.D., and Doig, A.J. (2003). Distinguishing enzyme structures from non-enzymes without alignments. J. Mol. Biol. *330*, 771–783.

112. Fout, A., Byrd, J., Shariat, B., and Ben-Hur, A. (2017). Protein interface prediction using graph convolutional networks. Adv. Neural Inf. Process. Syst. 6530–6539. https://papers.nips.cc/paper/7231-protein-interface-prediction-using-graph-convolutional-networks.

113. Zamora-Resendiz, R., and Crivelli, S. (2019). Structural learning of proteins using graph convolutional neural networks. bioRxiv, 610444. https://www.biorxiv.org/content/10.1101/610444v1.

114. Gligorijevic, V., Renfrew, P.D., Kosciolek, T., Leman, J.K., Cho, K., Vatanen, T., et al. (2019). Structure-based function prediction using graph convolutional networks. bioRxiv, 786236. https://www.biorxiv.org/content/10.1101/786236v2.

115. Torng, W., and Altman, R.B. (2019). Graph convolutional neural networks for predicting drug-target interactions. J. Chem. Inf. Model. *59*, 4131–4149.

116. Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., and Correia, B. (2020). Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nat. Methods *17*, 184–192.

117. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond Euclidean data. IEEE Signal. Process. Mag. *34*, 18–42.

118. Nerenberg, P.S., and Head-Gordon, T. (2018). New developments in force fields for biomolecular simulations. Curr. Opin. Struct. Biol. *49*, 129–138.

119. Derevyanko, G., Grudinin, S., Bengio, Y., and Lamoureux, G. (2018). Deep convolutional networks for quality assessment of protein folds. Bioinformatics *34*, 4046–4053.

120. Best, R.B., Zhu, X., Shim, J., Lopes, P.E., Mittal, J., Feig, M., and MacKerell, A.D., Jr. (2012). Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone φ, ψ and side-chain $\chi1$ and $\chi2$ dihedral angles. J. Chem. Theor. Comput. *8*, 3257–3273.

121. Weiner, S.J., Kollman, P.A., Case, D.A., Singh, U.C., Ghio, C., Alagona, G., Profeta, S., and Weiner, P. (1984). A new force field for molecular mechanical simulation of nucleic acids and proteins. J. Am. Chem. Soc. *106*, 765–784.

122. 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., et al. (2017). The Rosetta all-atom energy function for macromolecular modeling and design. J. Chem. Theor. Comput. *13*, 3031–3048.

123. Behler, J., and Parrinello, M. (2007). Generalized neural-network representation of high-dimensional potential-energy surfaces. Phys. Rev. Lett. *98*, 146401.

124. Smith, J.S., Isayev, O., and Roitberg, A.E. (2017). ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. Chem. Sci. *8*, 3192–3203.

125. Smith, J.S., Nebgen, B., Lubbers, N., Isayev, O., and Roitberg, A.E. (2018). Less is more: sampling chemical space with active learning. J. Chem. Phys. *148*, 241733.

126. Schütt, K.T., Arbabzadah, F., Chmiela, S., Müller, K.R., and Tkatchenko, A. (2017). Quantum-chemical insights from deep tensor neural networks. Nat. Commun. *8*, 1–8.

127. Schütt, K.T., Sauceda, H.E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. (2018). SchNet—a deep learning architecture for molecules and materials. J. Chem. Phys. *148*, 241722.

128. Zhang, L., Han, J., Wang, H., Car, R., and Weinan, E. (2018). Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. Phys. Rev. Lett. *120*, 143001.

129. Unke, O.T., and Meuwly, M. (2019). PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges. J. Chem. Theor. Comput. *15*, 3678–3693.

130. Zubatyuk, R., Smith, J.S., Leszczynski, J., and Isayev, O. (2019). Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network. Sci. Adv. *5*, eaav6490.

131. Lahey, S.-L.J., and Rowley, C.N. (2020). Simulating protein-ligand binding with neural network potentials. Chem. Sci. *11*, 2362–2368.

132. Wang, Z., Han, Y., Li, J., and He, X. (2020). Combining the fragmentation approach and neural network potential energy surfaces of fragments for accurate calculation of protein energy. J. Phys. Chem. B *124*, 3027–3035.

133. Senn, H.M., and Thiel, W. (2009). QM/MM methods for biomolecular systems. Angew. Chem. Int. Ed. *48*, 1198–1229.

134. Wang, Y., Fass, J., and Chodera, J.D. (2020). End-to-End Differentiable Molecular Mechanics Force Field Construction (arXiv). https://arxiv.org/abs/2010.01196.

135. Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A.E., and Kolinski, A. (2016). Coarse-grained protein models and their applications. Chem. Rev. *116*, 7898–7936.

136. Zhang, L., Han, J., Wang, H., Car, R., and Weinan, E. (2018). DeePCG: constructing coarse-grained models via deep neural networks. J. Chem. Phys. *149*, 034101.

137. Patra, T.K., Loeffler, T.D., Chan, H., Cherukara, M.J., Narayanan, B., and Sankaranarayanan, S.K. (2019). A coarse-grained deep neural network model for liquid water. Appl. Phys. Lett. *115*, 193101.

138. Wang, J., Olsson, S., Wehmeyer, C., Pérez, A., Charron, N.E., De Fabritiis, G., Noé, F., and Clementi, C. (2019). Machine learning of coarse-grained molecular dynamics force fields. ACS Cent. Sci. *5*, 755–767.

139. Wang, W., and Gómez-Bombarelli, R. (2019). Learning coarse-grained particle latent space with auto-encoders. Adv. Neural Inf. Process. Syst. *1*.

140. Li, Z., Wellawatte, G.P., Chakraborty, M., Gandhi, H.A., Xu, C., and White, A.D. (2020). Graph neural network based coarse-grained mapping prediction. Chem. Sci. *11*, 9524–9531.

141. Jones, D.T., Buchan, D.W., Cozzetto, D., and Pontil, M. (2011). PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics *28*, 184–190.

142. Di Lena, P., Nagata, K., and Baldi, P. (2012). Deep architectures for protein contact map prediction. Bioinformatics *28*, 2449–2457.

143. Eickholt, J., and Cheng, J. (2012). Predicting protein residue-residue contacts using deep networks and boosting. Bioinformatics *28*, 3066–3072.

144. Seemayer, S., Gruber, M., and Söding, J. (2014). CCMpred—fast and precise prediction of protein residue-residue contacts from correlated mutations. Bioinformatics *30*, 3128–3130.

145. Skwark, M.J., Raimondi, D., Michel, M., and Elofsson, A. (2014). Improved contact predictions using the recognition of protein like contact patterns. PLoS Comput. Biol. *10*, e1003889.

146. Jones, D.T., Singh, T., Kosciolek, T., and Tetchner, S. (2014). MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. Bioinformatics *31*, 999–1006.

147. Xu, J. (2019). Distance-based protein folding powered by deep learning. Proc. Natl. Acad. Sci. U S A *116*, 16856–16865.

148. Jones, D.T., and Kandathil, S.M. (2018). High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. Bioinformatics *34*, 3308–3315.

149. Hanson, J., Paliwal, K., Litfin, T., Yang, Y., and Zhou, Y. (2018). Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. Bioinformatics *34*, 4039–4045.

150. Kandathil, S.M., Greener, J.G., and Jones, D.T. (2019). Prediction of interresidue contacts with DeepMetaPSICOV in CASP13. Proteins *87*, 1092–1099.

151. Hou, J., Wu, T., Cao, R., and Cheng, J. (2019). Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. Proteins *87*, 1165–1178.

152. Zheng, W., Li, Y., Zhang, C., Pearce, R., Mortuza, S., and Zhang, Y. (2019). Deep-learning contact-map guided protein structure prediction in CASP13. Proteins *87*, 1149–1164.

153. Wu, Q., Peng, Z., Anishchenko, I., Cong, Q., Baker, D., and Yang, J. (2020). Protein contact prediction using metagenome sequence data and residual neural networks. Bioinformatics *36*, 41–48.

154. Marks, D.S., Colwell, L.J., Sheridan, R., Hopf, T.A., Pagnani, A., Zecchina, R., and Sander, C. (2011). Protein 3D structure computed from evolutionary sequence variation. PLoS One *6*, e28766.

155. Ma, J., Wang, S., Wang, Z., and Xu, J. (2015). Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. Bioinformatics *31*, 3506–3513.

156. Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat. Methods *9*, 173–175.

157. Fariselli, P., Olmea, O., Valencia, A., and Casadio, R. (2001). Prediction of contact maps with neural networks and correlated mutations. Protein Eng. *14*, 835–843.

158. Horner, D.S., Pirovano, W., and Pesole, G. (2007). Correlated substitution analysis and the prediction of amino acid structural contacts. Brief. Bioinform. *9*, 46–56.

159. Monastyrskyy, B., d'Andrea, D., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2014). Evaluation of residue–residue contact prediction in CASP10. Proteins *82*, 138–153.

160. Xu, J., and Wang, S. (2019). Analysis of distance-based protein structure prediction by deep learning in CASP13. Proteins *87*, 1069–1081.

161. Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2018). Critical assessment of methods of protein structure prediction (CASP)—Round XII. Proteins *86*, 7–15.

162. Wang, S., Li, W., Liu, S., and Xu, J. (2016). RaptorX-Property: a web server for protein structure property prediction. Nucleic Acids Res. *44*, W430–W435.

163. Gao, Y., Wang, S., Deng, M., and Xu, J. (2018). RaptorX-Angle: real-value prediction of protein backbone dihedral angles through a hybrid method of clustering and deep learning. BMC Bioinformatics *19*, 100.

164. AlQuraishi, M. (2019). AlphaFold at CASP13. Bioinformatics *35*, 4862–4865.

165. Zemla, A., Venclovas, Č., Moult, J., and Fidelis, K. (1999). Processing and analysis of CASP3 protein structure predictions. Proteins *37*, 22–29.

166. Kingma, D.P., Mohamed, S., Rezende, D.J., and Welling, M. (2014). Semi-supervised learning with deep generative models. Adv. Neural Inf. Process. Syst. 3581–3589.

167. Desmet, J., De Maeyer, M., Hazes, B., and Lasters, I. (1992). The dead-end elimination theorem and its use in protein side-chain positioning. Nature *356*, 539–542.

168. Krivov, G.G., Shapovalov, M.V., and Dunbrack, R.L. (2009). Improved prediction of protein side-chain conformations with SCWRL4. Proteins *77*, 778–795.

169. Liu, K., Sun, X., Ma, J., Zhou, Z., Dong, Q., Peng, S., Wu, J., Tan, S., Blobel, G., and Fan, J. (2017). Prediction of amino acid side chain conformation using a deep neural network. arXiv *1707*, 08381.

170. Du, Y., Meier, J., Ma, J., Fergus, R., and Rives, A. (2020). Energy-based models for atomic-resolution protein conformations. arXiv *2004*, 13167.

171. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A Tutorial on Energy-Based Learning (Predicting Structured Data), p. 1.

172. Zeng, H., Wang, S., Zhou, T., Zhao, F., Li, X., Wu, Q., and Xu, J. (2018). ComplexContact: a web server for inter-protein contact prediction using deep learning. Nucleic Acids Res. *46*, W432–W437.

173. Wang, S., Li, Z., Yu, Y., and Xu, J. (2017). Folding membrane proteins by deep transfer learning. Cell Syst. *5*, 202–211.e3.

174. Tsirigos, K.D., Peters, C., Shu, N., Käll, L., and Elofsson, A. (2015). The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. Nucleic Acids Res. *43*, W401–W407.

175. Alford, R.F., and Gray, J.J. (2020). Big data from sparse data: diverse scientific benchmarks reveal optimization imperatives for implicit membrane energy functions. Biophys. J. *118*, 361a.

176. Stein, A., and Kortemme, T. (2013). Improvements to robotics-inspired conformational sampling in Rosetta. PLoS One *8*, e63090.

177. Ruffolo, J.A., Guerra, C., Mahajan, S.P., Sulam, J., and Gray, J.J. (2020). Geometric potentials from deep learning improve prediction of CDR H3 loop structures. Bioinformatics *36*, i268–i275.

178. Nguyen, S.P., Li, Z., Xu, D., and Shang, Y. (2017). New deep learning methods for protein loop modeling. IEEE/ACM Trans. Comput. Biol. Bioinform. *16*, 596–606.

179. Li, Z.; Nguyen, S.P.; Xu, D.; Shang, Y. Protein loop modeling using deep generative adversarial network. Proceedings—International Conference on Tools with Artificial Intelligence, ICTAI. 2018; pp 1085–1091.

180. Porebski, B.T., and Buckle, A.M. (2016). Consensus protein design. Protein Eng. Des. Select. *29*, 245–251.

181. Killoran, N., Lee, L.J., Delong, A., Duvenaud, D., and Frey, B.J. (2017). Generating and designing DNA with deep generative models. arXiv *1712*, 06148.

182. Gupta, A., and Zou, J. (2018). Feedback GAN FBGAN for DNA: a novel feedback-loop architecture for optimizing protein functions. arXiv *1804*, 01694.

183. Brookes, D.H., Park, H., and Listgarten, J. (2019). Conditioning by adaptive sampling for robust design. arXiv *1901*, 10060.

184. Yu, C.-H., Qin, Z., Martin-Martinez, F.J., and Buehler, M.J. (2019). A self-consistent sonification method to translate amino acid sequences into musical compositions and application in protein design using artificial intelligence. ACS Nano *13*, 7471–7482.

185. Costello, Z., and Martin, H.G. (2019). How to hallucinate functional proteins. arXiv *1903*, 00458.

186. Chhibbar, P., and Joshi, A. (2019). Generating protein sequences from antibiotic resistance genes data using generative adversarial networks. arXiv *1904*, 13240.

187. Riesselman, A.J., Shin, J.-E., Kollasch, A.W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A.C., and Marks, D.S. (2019). Accelerating protein design using autoregressive generative models. bioRxiv, 757252.

188. Davidsen, K., Olson, B.J., DeWitt, W.S., III, Feng, J., Harkins, E., Bradley, P., and Matsen IV, F.A. (2019). Deep generative models for T cell receptor protein sequences. eLife *8*, https://doi.org/10.7554/eLife.46935.

189. Han, X., Zhang, L., Zhou, K., and Wang, X. (2019). ProGAN: protein solubility generative adversarial nets for data augmentation in DNN framework. Comput. Chem. Eng. *131*, 106533.

190. Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Zrimec, J., Poviloniene, S., et al. (2019). Expanding functional protein sequence space using generative adversarial networks. bioRxiv, 789719, https://doi.org/10.1101/789719. https://www.biorxiv.org/content/10.1101/789719v2.

191. Sabban, S., and Markovsky, M. (2020). RamaNet: computational de novo helical protein backbone design using a long short-term memory generative neural network. F1000Research *9*, 298.

192. Eguchi, R.R., Anand, N., Choe, C.A., and Huang, P.-S. (2020). Ig-VAE: generative modeling of immunoglobulin proteins by direct 3D coordinate generation. bioRxiv, 242347. https://www.biorxiv.org/content/10.1101/2020.08.07.242347v1.

193. Anishchenko, I., Chidyausiku, T.M., Ovchinnikov, S., Pellock, S.J., Baker, D., and Harvard, J. (2020). De novo protein design by deep network hallucination. bioRxiv, 211482. https://www.biorxiv.org/content/10.1101/2020.07.22.211482v1.

194. Wang, J., Cao, H., Zhang, J.Z., and Qi, Y. (2018). Computational protein design with deep learning neural networks. Sci. Rep. *8*, 6349.

195. Greener, J.G., Moffat, L., and Jones, D.T. (2018). Design of metalloproteins and novel protein folds using variational autoencoders. Sci. Rep. *8*, 1–12.

196. Chen, S., Sun, Z., Lin, L., Liu, Z., Liu, X., Chong, Y., Lu, Y., Zhao, H., and Yang, Y. (2019). To improve protein sequence profile prediction through image captioning on pairwise residue distance map. J. Chem. Inf. Model. *60*, 391–399.

197. Zhang, Y., Chen, Y., Wang, C., Lo, C.-C., Liu, X., Wu, W., and Zhang, J. (2019). ProDCoNN: protein design using a convolutional neural network. Proteins *88*, 819–829.

198. Shroff, R., Cole, A.W., Morrow, B.R., Diaz, D.J., Donnell, I., Gollihar, J., Ellington, A.D., and Thyer, R. (2019). A structure-based deep learning framework for protein engineering. bioRxiv, 833905.

199. Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., and Kim, P.M. (2019). Designing real novel proteins using deep graph neural networks. bioRxiv, 868935.

200. Karimi, M., Zhu, S., Cao, Y., and Shen, Y. (2019). De novo protein design for novel folds using guided conditional Wasserstein generative adversarial networks gcWGAN. bioRxiv, 769919.

201. Qi, Y., and Zhang, J.Z. (2020). DenseCPD: improving the accuracy of neural-network-based computational protein sequence design with DenseNet. J. Chem. Inf. Model. *60*, 1245–1252.

202. Anand, N., Eguchi, R.R., Derry, A., Altman, R.B., and Huang, P. (2020). Protein sequence design with a learned potential. bioRxiv, 895466.

203. Norn, C., Wicky, B.I., Juergens, D., Liu, S., Kim, D., Koepnick, B., et al. (2020). Protein sequence design by explicit energy landscape optimization. bioRxiv, 218917, https://doi.org/10.1101/2020.07.23.218917. https://www.biorxiv.org/content/10.1101/2020.07.23.218917v1.full.

204. Waghu, F.H., Gopi, L., Barai, R.S., Ramteke, P., Nizami, B., and Idicula-Thomas, S. (2014). CAMP: collection of sequences and structures of antimicrobial peptides. Nucleic Acids Res. *42*, D1154–D1158.

205. Grisoni, F., Neuhaus, C.S., Gabernet, G., Müller, A.T., Hiss, J.A., and Schneider, G. (2018). Designing anticancer peptides by constructive machine learning. ChemMedChem *13*, 1300–1302.

206. Yu, F., and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv *1511*, 07122.

207. Gupta, A., and Zou, J. (2019). Feedback GAN for DNA optimizes protein functions. Nat. Machine Intelligence *1*, 105–111.

208. Kuhlman, B., and Baker, D. (2000). Native protein sequences are close to optimal for their structures. Proc. Natl. Acad. Sci. U S A *97*, 10383–10388.

209. Li, Z., Yang, Y., Faraggi, E., Zhan, J., and Zhou, Y. (2014). Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. Proteins *82*, 2565–2573.

210. Karimi, M., Zhu, S., Cao, Y., and Shen, Y. (2020). De novo protein design for novel folds using guided conditional Wasserstein generative adversarial networks. J. Chem. Inf. Model. https://doi.org/10.1021/acs.jcim.0c00593.

211. Hou, J., Adhikari, B., and Cheng, J. (2017). DeepSF: deep convolutional neural network for mapping protein sequences to folds. Bioinformatics *34*, 1295–1303.

212. Jelinek, F., Mercer, R.L., Bahl, L.R., and Baker, J.K. (1977). Perplexity—a measure of the difficulty of speech recognition tasks. J. Acoust. Soc. Am. *62*, S63.

213. Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., and Kim, P. (2019). Fast and flexible design of novel proteins using graph neural networks. bioRxiv, 868935.

214. Ramachandran, G.N. (1963). Stereochemistry of polypeptide chain configurations. J. Mol. Biol. *7*, 95–99.

215. (2015). https://research.googleblog.com/2015/06/Ωinceptionism-going-deeper-into-neural.html.

216. Sutton, R.S., and Barto, A.G. (2018). Reinforcement Learning: An Introduction (MIT press).

217. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition 2009, 248–255.

218. Mayr, A., Klambauer, G., Unterthiner, T., and Hochreiter, S. (2016). DeepTox: toxicity prediction using deep learning. Front. Environ. Sci. *3*, 80.

219. Brown, N., Fiscato, M., Segler, M.H., and Vaucher, A.C. (2019). GuacaMol: benchmarking models for de novo molecular design. J. Chem. Inf. Model. *59*, 1096–1108.

220. Lutter, M., Ritter, C., and Peters, J. (2019). Deep Lagrangian networks: using physics as model prior for deep learning. arXiv *1907*, 04490.

221. Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. Adv. Neural Inf. Process. Syst. 15379–15389.

222. Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. *378*, 686–707.

223. Zepeda-Núñez, L., Chen, Y., Zhang, J., Jia, W., Zhang, L., and Lin, L. (2019). Deep Density: circumventing the Kohn-Sham equations via symmetry preserving neural networks. arXiv *1912*, 00775.

224. Han, J., Li, Y., Lin, L., Lu, J., Zhang, J., and Zhang, L. (2019). Universal approximation of symmetric and anti-symmetric functions. arXiv *1912*, 01765.

225. Shapovalov, M.V., and Dunbrack, R.L., Jr. (2011). A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. Structure *19*, 844–858.

226. Hintze, B.J., Lewis, S.M., Richardson, J.S., and Richardson, D.C. (2016). Molprobity's ultimate rotamer-library distributions for model validation. Proteins *84*, 1177–1189.

227. Jensen, K.F., Coley, C.W., and Eyke, N.S. (2019). Autonomous discovery in the chemical sciences part I: progress. Angew. Chem. Int. Ed. *59*, 2–38.

228. Coley, C.W., Eyke, N.S., and Jensen, K.F. (2019). Autonomous discovery in the chemical sciences part II: outlook. Angew. Chem. Int. Ed. *59*, 2–25.

229. Coley, C.W., Thomas, D.A., Lummiss, J.A., Jaworski, J.N., Breen, C.P., Schultz, V., Hart, T., Fishman, J.S., Rogers, L., Gao, H., et al. (2019). A

robotic platform for flow synthesis of organic compounds informed by AI planning. Science *365*, eaax1566.

230. Barrett, R.; White, A.D. Iterative peptide modeling with active learning and meta-learning. arXiv preprint 2019, 1911.09103.

231. You, J., Liu, B., Ying, R., Pande, V., and Leskovec, J. (2018). Graph convolutional policy network for goal-directed molecular graph generation. Adv. Neural Inf. Process. Syst. 6410–6421.

232. Zhou, Z., Kearnes, S., Li, L., Zare, R.N., and Riley, P. (2019). Optimization of molecules via deep reinforcement learning. Sci. Rep. *9*, 1–10.

233. Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Bae, S., et al. (2004). Chip placement with deep reinforcement learning. arXiv *2020*, 10746.

234. Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z., and Foldit, P. (2010). Predicting protein structures with a multiplayer online game. Nature *466*, 756–760.

235. Koepnick, B., Flatten, J., Husain, T., Ford, A., Silva, D.-A., Bick, M.J., Bauer, A., Liu, G., Ishida, Y., Boykov, A., et al. (2019). De novo protein design by citizen scientists. Nature *570*, 390–394.

236. Czibula, G., Bocicor, M.-I., and Czibula, I.-G. (2011). A reinforcement learning model for solving the folding problem. Int. J. Comput. Technol. Appl. *2*, 171–182.

237. Jafari, R., and Javidi, M.M. (2020). Solving the protein folding problem in hydrophobic-polar model using deep reinforcement learning. SN Appl. Sci. *2*, 259.

238. Gao, W. (2020). Development of a Protein Folding Environment for Reinforcement Learning, M.Sc. thesis (Johns Hopkins University).

239. Angermueller, C., Dohan, D., Belanger, D., Deshpande, R., Murphy, K., and Colwell, L. (2020). Model-Based Reinforcement Learning for Biological Sequence Design (ICLR 2020 Conference). https://openreview.net/forum?id=HklxbgBKvr.

240. Zeiler, M.D., and Fergus, R. (2014). Visualizing and understanding convolutional networks. Eur. Conf. Comput. Vis. 818–833.

241. Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). SmoothGrad: removing noise by adding noise. arXiv *1706*, 03825.

242. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. Proceedings of the 34th International Conference on Machine Learning2017, 70, 3319–3328.

243. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. Adv. Neural Inf. Process. Syst. 9505–9515.

244. Shrikumar, A., Greenside, P., and Kundaje, A. (1704). Learning important features through propagating activation differences. arXiv *2017*, 02685.

245. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. Proceedings of the 31st International Conference on Neural Information Processing Systems 2017, 4768–4777.

246. Hannon, G.J. (2002). RNA interference. Nature *418*, 244–251.

247. Zhang, P., Woen, S., Wang, T., Liau, B., Zhao, S., Chen, C., Yang, Y., Song, Z., Wormald, M.R., Yu, C., et al. (2016). Challenges of glycosylation analysis and control: an integrated approach to producing optimal and consistent therapeutic drugs. Drug Discov. Today *21*, 740–765.

248. Sanchez-Lengeling, B., and Aspuru-Guzik, A. (2018). Inverse molecular design using machine learning: generative models for matter engineering. Science *361*, 360–365.

249. Coley, C.W., Jin, W., Rogers, L., Jamison, T.F., Jaakkola, T.S., Green, W.H., Barzilay, R., and Jensen, K.F. (2019). A graph-convolutional neural network model for the prediction of chemical reactivity. Chem. Sci. *10*, 370–377.

250. Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. (2019). Analyzing learned molecular representations for property prediction. J. Chem. Inf. Model. *59*, 3370–3388.

251. Gao, W., and Coley, C.W. (2020). The synthesizability of molecules proposed by generative models. J. Chem. Inf. Model. https://doi.org/10.1021/acs.jcim.0c00174.

252. Langan, R.A., Boyken, S.E., Ng, A.H., Samson, J.A., Dods, G., Westbrook, A.M., Nguyen, T.H., Lajoie, M.J., Chen, Z., Berger, S., et al. (2019). De novo design of bioactive protein switches. Nature *572*, 205–210.