Predicting the Real-Valued Inter-Residue Distances for Proteins

*Wenze Ding and Haipeng Gong**

Predicting protein structure from the amino acid sequence has been a challenge with theoretical and practical significance in biophysics. Despite the recent progresses elicited by improved inter-residue contact prediction, contact-based structure prediction has gradually reached the performance ceiling. New methods have been proposed to predict the inter-residue distance, but unanimously by simplifying the real-valued distance prediction into a multiclass classification problem. Here, a lightweight regression-based distance prediction method is shown, which adopts the generative adversarial network to capture the delicate geometric relationship between residue pairs and thus could predict the continuous, real-valued inter-residue distance rapidly and satisfactorily. The predicted residue distance map allows quick structure modeling by the CNS suite, and the constructed models approach the same level of quality as the other state-of-the-art protein structure prediction methods when tested on CASP13 targets. Moreover, this method can be used directly for the structure prediction of membrane proteins without transfer learning.

1. Introduction

Proteins participate in nearly all kinds of physiological activities and their 3D structures are essential for the functions. Since the finding that the protein structures are prescribed by their amino acid sequences, exploration of the relationship among protein sequence, structure, and function has been one of the core problems of molecular biophysics. Unlike experimental structure determination methods that are costly and technically prohibitive, predicting the protein structure via computational approaches could be applied in a high-throughput manner and thus is widely needed in practical applications ranging from protein design to pharmaceutical development.[1]

In traditional de novo protein structure prediction, the native structure is located by exhaustively searching the protein conformational space, using molecular dynamics simulations that employ empirical force fields or fragment-assembly-based/threading-assembly-based Monte Carlo simulations that use experimentally determined structures as templates and force fragments of the target protein to adopt their conformations.[2] Despite the successes, traditional methods become less powerful for hard protein targets that have complex topologies but limited homology to known structures, for example, the free-modeling (FM) targets in the critical assessment of protein structure prediction (CASP) competitions. In addition, these methods are computationally expensive in general, because the protein conformational space frequently has intimidatingly high dimensionality.

Breakthrough in the accuracy of protein structure prediction was observed in CASP11 and CASP12[3] (hosted in 2014 and 2016, respectively), which was mainly driven by the use of co-evolution information and deep learning algorithms. For a target protein, evolutionary couplings between residues could be detected and extracted from the multiple sequence alignment (MSA) to predict the binary contact matrix, assuming that spatial neighborhood of residues (so-called residue contact, strictly defined as the distance of C_β atoms ≤ 8 Å according to the CASP convention) would elicit correlated mutations over long evolutionary time. The contact matrix contains geometric constraining information that could be used by protein folding programs such as CONFOLD[4] to restore the atomic coordinates. On the other hand, traditional methods also benefit substantially from the integration of contact prediction in structure selection and energy evaluation.[5] Extraction of contact information from the MSA is a typical pattern recognition problem that is particularly suitable to be handled by deep learning techniques like convolution neural network,[6] because of the power of such techniques to identify the correlation between contacting residue pairs located far away in the contact map. Among many deep-learning-based approaches, RaptorX-Contact that adopts deep residual network (ResNet) outperforms others and makes huge influences in this field.[7]

W. Ding, Dr. H. Gong
MOE Key Laboratory of Bioinformatics
School of Life Sciences
Tsinghua University
Beijing 100084, China
E-mail: hgong@tsinghua.edu.cn
W. Ding, Dr. H. Gong
Beijing Advanced Innovation Center for Structural Biology
Tsinghua University
Beijing 100084, China

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/advs.202001314

DOI: 10.1002/advs.202001314

Nevertheless, contact prediction is just a compromise when accurate distance prediction is not available. Distance prediction has many intrinsic advantages over contact prediction for protein folding. First of all, contact prediction is a binary classification problem with unbalanced positive and negative samples (e.g., roughly 1 contact: 50 non-contacts for long-range residue pairs with sequence separation $\geq$ 24),[8] which frequently requires undersampling of negative ones during model training and thus may lead to the inconsistence between the prediction score and the real contacting probability for a residue pair. Therefore, contact-assisted protein folding methods usually only adopt a small part of predicted contacts with top scores for structure modeling, which is susceptible to the noises raised by very few wrongly predicted contacts. Direct prediction of distance map (i.e., the 2D matrix listing real-valued distances between all residue pairs) would avoid this problem, because all predicted values within a suitable interval (e.g., 4–16 Å) can be utilized and thus the disturbance introduced by the prediction errors of individual residue pairs may mitigate according to the law of large number. More importantly, distance matrices contain much more detailed information of protein structure than contact matrices, which could reduce conformational sampling more effectively and thus fold the protein more accurately and rapidly. Consequently, despite the great progresses introduced by contact prediction, first-ranked groups in the field of protein structure prediction like AlphaFold and RaptorX-Contact have switched their attention to distance prediction in CASP13[7,9] (hosted in 2018). Particularly, the success of AlphaFold in CASP13 has been mainly attributed to the more accurate prediction of inter-residue distances.[10]

Ideally, during the switch from contact prediction to distance prediction, the nature of the explored task should transit from a classification problem to a regression problem, because residue contacts are actually human-defined zero-one labels while distances are real-valued physical metrics. However, both AlphaFold and RaptorX-Contact simply chose to modify binary classification to multiclass classification. Instead of real-valued distances, they used a discrete representation with several fixed-width bins.[7,9] The rationale for their choices is mainly threefold. First and foremost, traditional regression loss functions used in deep neural network (DNN) like mean absolute error (MAE, also called L1 loss) and mean square error (also called L2 loss) measure the globally averaged deviation of the prediction from the ground truth. After loss minimization by DNN, the predicted distances may be pretty good on average but still far from satisfaction as individuals, which is of limited usefulness for protein folding. Generally, it is hard and needs many manual efforts to design effective losses for separate, special-purpose machinery. Second, modern DNN training procedures always add batch normalization (BN) layers to solve the problem of gradient vanishing or explosion, which normalize the forward-passing data into a standard normal distribution. Thus, without ingenious, human-designed mapping functions, common activation functions used in DNN are powerless of outputting positive real numbers like distances. Third, with their powerful, well-tested contact prediction networks in hand, these groups can conveniently use transfer learning techniques to get satisfactory distance prediction results.

In this work, we adopted the generative adversarial networks (GANs) to directly predict the real-valued inter-residue distances for proteins. GAN is a computer vision technique, containing a generator to produce outputs and a discriminator to classify outputs of the generator from the real ones (ground truths). With joint training, the generator would not only fit the pixel distribution of real image globally, but also highlight "important" pixel areas with sharp, precise values to fool the discriminator simultaneously.[11] In 2018, Anand and Huang applied the GANs to generate the distances between C_α atoms and then used the generated distance map to complete small structural corruptions for protein design problem.[12] Despite the interesting finding that the generator in GANs can effectively capture features of secondary structure elements, their model failed to generate distance maps of arbitrary size and could not be used to predict the structure of a given protein sequence. In this work, we solved all the obstructions for distance prediction via GANs and were able to predict the real-valued inter-residue distances for practical proteins with satisfactory accuracy for the first time. Other contributions of our work include 1) introducing new, effective data augmentation methods to produce more robust models, especially the augmentation of distance labels by molecular dynamics simulations, which considers the structural dynamics of proteins that are typically ignored in structural bioinformatics, 2) designing reversible mapping functions between positive real numbers and the interval of [−1, 1] to enable the direct training of DNN for continuous inter-residue distance regression, and 3) analyzing the effects of several technical choices and then summarizing some empirical laws for the deep learning solution of inter-residue distance prediction for proteins. When pipelined with the same protein folding program CNS suite,[13] structure models generated using our distance constraints are significantly better than those produced with the contact constraints from the state-of-the-art contact predictors like RaptorX-Contact[7] and TripletRes.[14] Moreover, when tested on available CASP13 targets, our structure models approach at least the same level of quality as the top protein structure prediction groups, including AlphaFold (A7D),[9] QUARK,[15] Zhang-Sever,[15] and RaptorX-DeepModeller.[7] Although trained mainly by protoplasmic soluble proteins, the generalizability of our predictor renders its application for the structure prediction of membrane proteins without the requirement of any transfer learning processes.

2. Results

2.1. A Preliminary GAN Model for Protein Distance Prediction

We first developed mapping functions to allow the back and forth transformation between real-valued distances or features and numbers in [−1, 1] (see Section 4 for details), through which the ground-truth distance maps could be converted to the interval of [−1, 1] to simplify the training of DNN models with BN layers and the prediction results could also be restored to the domain of real distances instantly. Particularly, the mapping functions were designed to have large gradients for distances between 4 and 16 Å, the range possessing rich information for the protein structure modeling. We then adopted the conditional GAN (cGAN) for protein inter-residue distance prediction. Similar to but distinct from primitive GANs, cGAN learns a generative model (generator, referred as G) that can generate the corresponding output

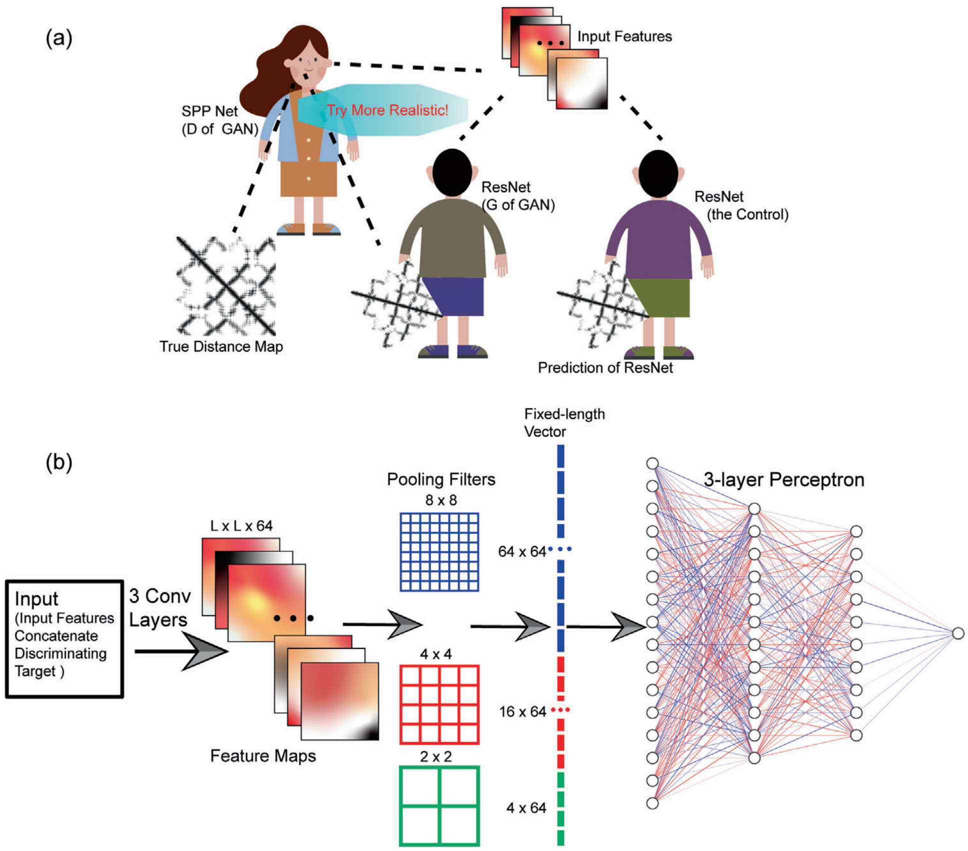

Figure 1. Schematic illustrations of the preliminary experiment. a) D and G of our GAN model contest as adversaries to improve the quality of predicted distance map, in comparison with the traditional ResNet as control. The dashed lines represent information flows. b) The architecture of the D model.

of expected size in the condition of an input.[11a] Here, a 40-layer ResNet, one of the most successful network architectures in this field,[7] was chosen as the generator of our cGAN and was also taken as the control to evaluate the performance gain of GANs over pure generative models (**Figure 1**a). The discriminator of our cGAN, referred as D, is trained to detect the outputs of G as "fake" from "real" under the condition of input features fed to G, whereas G tries to learn from the decision of D and produces indistinguishable outputs to "fool" D through the adversarial training procedure. More specifically, the loss function of D can be defined as a standard cross-entropy function for a binary classifier with a sigmoid output:

$$\mathrm{LossD} = -\mathrm{LossGAN}(\mathrm{G},\mathrm{D}) = -(E_{x,y}[\log \mathrm{D}(x,y)] + E_{x,z}[\log(1-\mathrm{D}(x,\mathrm{G}(x,z)))]) \tag{1}$$

where x, y, and z represent input features, real distance maps, and input noises, respectively, and E denotes expectation. Notably, unlike the common GANs that apply noises to ensure the randomness of outputs, we did not apply noises in G, because G in our experiments is a deterministic model to produce distance map given input sequence features. D tries to minimize Equation (1) against the adversarial G, which in return tries to maximize it (i.e., minimize its negative number, LossGAN). Besides fooling D, G should also constrain its outputs near the ground truths. Hence, it would be beneficial to combine a more traditional regression loss (RegLoss), and the final G loss is defined as

$$\mathrm{LossG} = \mathrm{LossGAN}(\mathrm{G},\mathrm{D}) + \lambda \times \mathrm{RegLoss} \tag{2}$$

where λ is a weight parameter to adjust the relative importance of two parts. In this section, we chose L1 loss as the regression loss and 258 for its weight. For the consistence between cGAN and the control, L1 loss of the control ResNet was also multiplied by the same λ:

$$\mathrm{LossControl} = \lambda \times \mathrm{RegLoss} \tag{3}$$

We set each individual protein as a mini-batch during training. For G of our cGAN and the control ResNet, 64 3 × 3 2D convolution filters with stride 1 and zero-padding "same" were adopted for each layer, followed by the leaky rectified linear unit (leaky-ReLU) and BN. For D, we concatenated the input features of G and the discriminating targets (i.e., outputs of G or ground truths) as its input. To solve the problem of variable sizes of individual proteins, we adopted spatial pyramid pooling[16] following the 3 convolutional layers (each with 64 3 × 3 2D filters) in D, where the max-pooling results of three different separations (8 × 8, 4 × 4, and 2 × 2 patches) of the feature map were concatenated as a fixed-length vector and were fed into a 3-layer perceptron

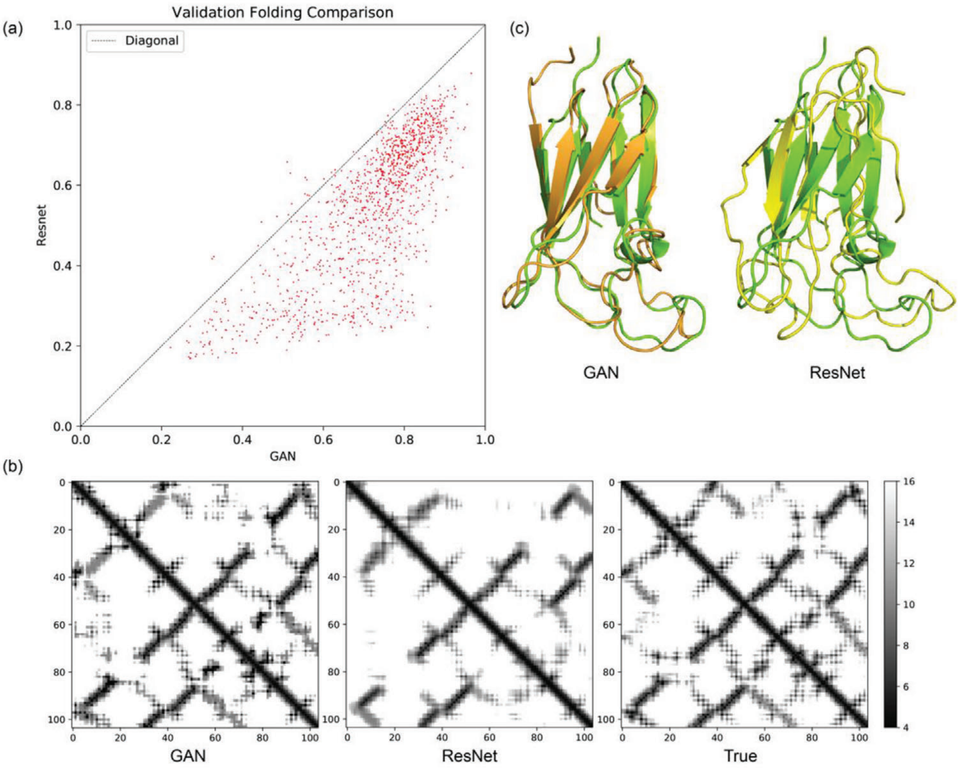

Figure 2. Comparison between our cGAN system and ResNet (the control) for protein structure prediction. a) Comparison of the TM-scores of the structure models produced by CNS-based folding using the distance predictions of cGAN and ResNet as restraints in the validation set. b) Predicted distance maps by cGAN (left) and ResNet (middle) are compared with the true distance map (right) for a case target (PDB ID: 2II8). The bar on the right indicates the grayscale for the predicted distance (Å). c) Structure alignment of the best folded models using cGAN (orange, left) and ResNet (yellow, right) predictions against the crystal structure (green).

to output the probability of the given distance map to be true (Figure 1b). The training procedures of cGAN and the control ResNet were completely the same, using the Adam optimizer for 100 epochs with the learning rate set as 1e-4, 1e-5, and 1e-6 for the first 20, the middle 30, and the last 50 epochs, respectively. We randomly chose 5642 proteins from the dataset of 6862 chains as the training set, and left the rest 1220 proteins as the validation set. To speed up training, the maximal length of proteins was limited to 400 residues.

Table S1, Supporting Information, summaries the prediction errors by the cGAN and the control ResNet in the validation set. For residue pairs with the predicted distances falling between 4 and 16 Å (the range having rich information for protein structure modeling), ResNet seems to reach lower prediction error than cGAN on average (1.832 Å vs 1.938 Å). However, are the "seemingly better" results by ResNet really benefiting the protein structure modeling? To address this question, we collected all predicted distances within 4 and 16 Å to construct the distance constraining matrix and invoked the CNS suite[13] (using a similar protocol to CONFOLD[4]) to fold the proteins in the validation set. To ensure that CNS suite indeed uses the predicted residue distances for structure modeling, we chose a narrow distance range of ±0.4 Å around the predicted value. Quality of the top 1 models was evaluated by TM-score. As shown in **Figure 2**a, cGAN defeats the control ResNet for most targets in the validation set. The models folded by cGAN predictions reach an average TM-score of 0.722, with 92.7% of the targets folded in the correct topology (i.e., TM-score > 0.5). In contrast, the average TM-score of ResNet-based folding is 0.544, with only 63.9% of the targets folded correctly. Hence, despite the slight weakening of the overall distance prediction accuracy, introduction of the GAN loss that comprehensively considers the adversarial generator and discriminator (see Equations (1) and (2)) indeed improves the structure modeling based on the predicted distances.

Figure 2b shows the distance maps predicted by cGAN and ResNet as well as the ground truth for an example target (PDB ID: 2II8). Clearly, the prediction by ResNet is blurry overall, although locations and average values of the main stripes are roughly correct. In contrast, despite many tiny mistakes, the prediction by cGAN contains much more details with sharp edges. The sharp contrasts between pixel signals captured by cGAN prediction describe the subtle correlations between individually predicted distances, which imply the delicate geometric relationship between residue pairs. Consequently, the structure model generated by cGAN prediction agrees with the native structure significantly better than that by ResNet prediction (Figure 2c).

The fitting power of ResNet is guaranteed by the multiple stacking of residual blocks even when the size of convolution filter is small, as the receptive field would be amplified in a cascaded way and thus the interdependency of two arbitrary residue pairs could be captured. Thus, the question is focused on what we want our neural network to fit, and since the network learns to minimize a loss function that evaluates how close the outputs of the network and our desires are, the question finally becomes how we define the loss of our neural network. It is well known that traditional regression loss like L1 or L2 losses could capture the low-frequencies, that is, average information, accurately from inputs. They measure the global quality of outputs and thus drive the networks to produce predicting values around the local average, which as a result may blur their outputs. However, these accurate low-frequencies are far from the demand of practical usage in protein folding, and what we really want is a realistic residue distance map with sharp contrasts between pixels. Designing effective losses specifically for the extraction of these high-frequencies, that is, texture information, is difficult because the high frequencies somehow represent the general properties of polypeptides or the protein folding mechanism like the interacting pattern between secondary structure elements and the local folding propensity of loop regions. Avoiding directly defining such kind of texture losses, our GAN solved this problem through achieving a high-level goal of "producing reality-indistinguishable predictions" and used a neural network D to learn this loss. At the same time, our GAN trained its generative model G to minimize the learned loss, which successfully suppressed the unrealistic blurs and reproduced high-frequencies.

2.2. Introducing Patch Classifiers to the Architecture of D

As a data augmentation method, cropping has been proved as useful in practice by many research groups in this field. For example, AlphaFold randomly chose 64 × 64 patches from the protein feature map when training their 660-layer ResNet, which brought about many benefits, such as solving inconsistency problem of protein length variation, helping distributed training, avoiding overfitting problem and facilitating ensemble average for inference.[9] However, direct imitation of such cropping in our case failed in the GAN training.

Markovian discriminator was proposed recently to model high-frequencies in GAN.[11e] Instead of determining whether the entire output is "real" or not, such kind of discriminators pay attention to subtle structure differences in output patches of fixed size. Inspired by this idea, we implemented our patch classifier as an alternative cropping method in D through a fully convolutional network (FCN). Each layer of this FCN adopts the 4 × 4 convolution kernel with the leaky-ReLU set as activation and zeros padded around inputs when necessary. The kernel stride of precedent layers was set as 2 to enlarge receptive field rapidly while the stride of the last two layers was set as 1 to better integrate information captured in each neuron. The channel number of the first convolution layer was set as 128, and the following ones were doubled at each turn except the last layer, where the channel number was set to 1. Sigmoid function was used as activation for the last convolution layer to output the probability of the corresponding patch to be true. The patch size could be modified with the depth variation of this FCN. For example, as shown in Figure S1, Supporting Information, if we want the classifier to focus on 34 × 34 patches, the FCN should have 4 layers totally, with strides of (2, 2, 1, 1) and channel numbers of (128, 256, 512, 1). Notably, for each target protein, we applied dense sampling of patches to ensure coverage of the whole distance map without omission.

We observed that the patch classifiers that have fewer parameters and faster speed indeed produced better results than the single classifier that makes judgement on the entire input of distance map (Figure S2, Supporting Information). Because residue pairs separated by a patch diameter or longer intervals are frequently independent considering the statistical length of protein secondary structure elements (i.e., helices and sheets), patch classifiers could model the residue distance map as a Markov random field. Thus, the loss learned by patch classifiers should be useful for the extraction of the special texture pattern of distance map.

The characteristic of distance map hints us that we should pay more attention to those patches having stripes of strong signals (i.e., predicted distance between 4–16 Å) since only such predictions are meaningful and contributive to protein folding in our case. However, the distance map is usually dominated by blank background regions (i.e., predicted distance > 16 Å), which albeit lacking useful information are highly likely to be judged by D as "real" because they seem identical to the corresponding blank patches on the ground-truth map (Figure S3, Supporting Information). To reduce such confusion of D, we modified its cross-entropy loss from Equation (1) to

$$\begin{aligned}\mathrm{LossD} = &-(E_{x,y}[\log \mathrm{D}(x, y)] \\ &+ E_x[\log(\mathrm{CLIP}(0.9 - \mathrm{D}(x, \mathrm{G}(x)), 0, 0.9))])\end{aligned} \tag{4}$$

where CLIP(fun(a),0,0.9) is a clip function that only retains values of fun(a) within the 0–0.9 interval. Through this modification, patches seeming realistic (e.g., those blank background regions without stripes) at the first beginning when G has not learned useful information would be filtered out for the decision of D.

2.3. Optimizing the Generative Model G

As G is the major undertaker for information integration and extraction from inputs and the actually used part during inference, its architecture is vital to performance of the overall network. In this section, we adjusted all components of G one after another and summarized some empirical laws of the technical choices for distance prediction. All effects of adjustments were analyzed by fivefold cross validation on the protein dataset of 6862 chains.

During training, we frequently observed the premature convergence of the GAN loss of G (LossGAN(G,D) in Equation (2)). This is because D is likely to reject all the distance maps produced by G with high confidence after a few epochs when G cannot really learn something, considering that the regression task of G is much harder than the classification task of D. Albeit correct, such rejections quickly reduce the loss of D (Equation (1)), which prevents the further learning by G. To solve it, we modified the GAN loss of G to favor the cases when D accepts distance maps predicted by G as "real" ones:

$$\mathrm{LossG} = E_x[-\log \mathrm{D}(x, \mathrm{G}(x))] + \lambda \times \mathrm{RegLoss} \tag{5}$$

We tried a number of common network architectures for G, including different ResNet variants, DenseNet, and U-Net

(Table S2, Supporting Information). Among them, U-Net was hard to implement for inputs of variable sizes by available convolutional depression and recovery techniques, and we had to pad zeros around the inputs to ensure the length uniformity, which impaired the performance, because the padded zone might be much larger than original size for many proteins. Taking computational consumption (i.e., amount of parameters and floating-point operations per second (FLOPs)) into account, the performance of DenseNet is not satisfactory. For ResNet, the 3-layer-per-block variant outperforms the 2-layer-per-block one. Bottleneck structure seems not beneficial to distance prediction because the 1 × 1 kernel is incapable of enlarging the receptive field. We finally picked up the ResNet architecture and augmented it under the guidance of EfficientNet[17] to generate two separate models for proteins of small/medium and large sizes, respectively (see the next section for details).

As for the convolution kernel of G, we tried different kernel sizes, kernels with dilation and separable kernels (Table S3, Supporting Information). Larger kernels have better performance by considering more complex interdependencies. As a frequently used technique in contact prediction and distance prediction with multi-classifiers, dilated kernels underperform normal ones. The reason of this phenomenon is that unlike classification problem that only needs to learn categorical information, real-valued distance regression requires large receptive field without any omission, especially for the extraction of texture information. We finally chose the 7 × 7 kernel. Since the training of models with 7 × 7 kernels was relative slow, we adopted parameter sharing technique proposed in the work of ShaResNet[18] to reduce the amount of parameters and accelerate training. Unfortunately, this procedure led to training failure.

In the evaluation of activation functions, the Swish function

$$\mathrm{Swish}(x) = x \times \mathrm{sigmoid}(\beta x) \tag{6}$$

which has a learnable parameter β, achieves the best performance (Table S4, Supporting Information). Generally, activations with tunable parameters outperform those without, because they can mimic real biological neural networks, in which every individual neuron has its own property and activation threshold. It is noteworthy that random-ReLU (R-ReLU) impairs the performance, which is inconsistent with our previous experience on protein contact prediction. Among the regression losses tested, the MAE loss (or L1 loss) is the simplest but the most effective one (Table S5, Supporting Information). This is because instead of biasing predictions of larger ground-truth values that usually have larger errors, the L1 loss balances various kinds of predictions well and thus performs more robustly.

Among features from various sources, 2D features are the most valuable. However, unlike AlphaFold that uses enormous 2D features directly from the Potts model to ensure information coverage,[9] 2D features in our model only occupy 3.07% of the inputs (4 out of 130, see Experimental Section). Although these features are extracted from the MSA and are thus informative, their contributions may be submerged by the large amount of redundant information produced by the broadcasting of 1D features. Besides, the unbalanced value distribution of the protein distance map (e.g., the sparse distribution of intense “stripes” on the vast blank background area) further complicates the training. Inspired by these, attention aiming to reweight the channels (i.e., different features) and pixels (i.e., different regions) of the input features in consideration of individual targets is necessary in our system. We implemented an attention module with global average and max pooling (see Section 4) and the results supported its effectiveness in improving the model performance (Table S6, Supporting Information). In the validation set, the channel-wised attention sufficiently suppresses the weights of redundant information from the broadcasting of 1D features (Figure S4, Supporting Information). Meanwhile, the pixel-wised attention effectively adjusts the weights of individual pixels to facilitate information extraction (Figure S5, Supporting Information).

2.4. Data Augmentation with Biological Significance

What kind of structures should the predictors in the bioinformatics field predict? It is an open question since the structures from the protein data bank (PDB) that are used as ground truths for model training are static structures determined in non-physiological conditions. Different crystallization situations, different structure analysis technologies (NMR, X-Ray, cryo-EM, etc.) and even different structure computation methods may lead to structure variation. More importantly, these static structures are unable to reflect the dynamic behaviors of real proteins in aqueous environments. Molecular dynamics (MD) simulations could solve this problem, because physiological environments are constructed and empirical force fields are adopted to observe the protein dynamics starting from the PDB structure in these simulations. To guarantee the generalizability and robustness of our predictor as well as to consider the protein dynamics, we augmented data via MD simulations (see Section 4). For each protein in our training set, 500 structures were collected with even separation from the 5-nanosecond trajectory of equilibrium simulation. The conformational change reaches the RMSD level of 6 Å on average at the end of the simulation (Figure S6, Supporting Information), which ensures that structural dynamics are sufficiently considered by our data augmentation. To validate the contribution of this data augmentation on practical protein structure prediction, we trained two models of the same architecture (L1 loss weight $\lambda = 100$, patch size of D = 70 and without clipping) using structures from PDB (referred as Model 1) and structures produced by MD simulations (referred as Model 2), respectively, and utilized their results to fold proteins in the CASP13 set and non-redundant membrane protein set by the CNS suite. Enhancement in the quality of folded structures by Model 2 supports that our data augmentation indeed captures something with biological significance, which improves the generalizability of distance prediction and benefits the distance-based structure modeling (Table S7, Supporting Information).

We trained our final models at L1 loss weight $\lambda = 158$, patch size of D = 70 and clipping of the D loss. During training, the maximally allowed protein length and the model size are a pair of mutually constrained parameters in a fixed environment (TITAN RTX with 24 190 MiB memory). For instance, model complexity and thus prediction accuracy will be sacrificed when relaxing the protein length restriction in training. However, many protein domains, not only in our training set but also in academic research and practical usage of protein structure prediction, are no longer

Table 1. Comparison of the folding capability by our distance predictor against contact predictors in the CASP13 set.

	Average TM-score
TripletRes	0.568
RaptorX-Contact	0.527
Our GAN system	0.712

than 300 residues (Figure S7, Supporting Information). Under such consideration, we trained two separate models for different purposes. A high-capacity model of 77 convolution layers (referred as Model X) was trained using proteins of ≤ 300 residues to specifically process the distance prediction for small protein domains (length ≤ 350 residues), whereas a low-capacity model of 52 convolution layers (referred as Model L) was optimized with protein length restriction relaxed to 450 residues to undertake the prediction task for the other proteins (length > 350 residues). The combination of these two models compose a lightweight comprehensive model for inference, which optimally makes use of the limited computational resources without sacrificing the prediction accuracy for small targets. For each protein, we randomly chose 6 structures as ground truths from the available structures (one PDB structure plus all simulation-produced ones) during the training of each model. To stabilize the training of our GAN system, we used exponential moving average for loss when updating network parameters. We computed the average of the upper and lower triangles of outputs as the final distance prediction for each residue pair and then applied ±0.4 Å around the predicted value as the boundary of the allowed distance range when folding the protein by the CNS suite.

2.5. Evaluation

The usefulness of our distance prediction method was mainly evaluated in practical protein structure prediction. To validate the superiority of distance prediction to contact prediction, we compared our distance predictor with our contact predictor DeepConPred2[8] in the CASP12 set and with the top CASP13 contact predictors including RaptorX-Contact and TripletRes in the CASP13 set, by uniformly using the CNS suite to fold the proteins. The allowed distance range of each residue pair was set strictly and narrowly as the predicted value ±0.4 Å, but was set to [3.5, 8] for all contact predictors following the CONFOLD protocol (using top $2L$ contacts as CONFOLD suggests, where L is the protein length). Clearly, our GAN system significantly outperforms DeepConPred2 that has similar input features (Table S8, Supporting Information). Moreover, structure models constructed from our distance prediction have remarkably better quality than those generated based on the results of the state-of-the-art contact predictors (**Table 1**). The lead of our method by a large margin in TM-score supports the important contribution of real-valued distance prediction in protein structure prediction.

We also compared the structures folded using our distance prediction against the CASP13 top 3 protein structure prediction servers (QUARK, Zhang-Server, and RaptorX-DeepModeller) and the best human group (A7D, also known as AlphaFold) on 42 available CASP13 targets (38 for A7D, with four server-only targets T0950-D1, T0951-D1, T0967-D1, and T0971-D1 excluded due to the lack of A7D results). As shown in Table S9, Supporting Information, our method reaches an average TM-score of 0.712 for all targets, and the average TM-scores for FM targets and template-based-modeling (TBM) targets are 0.620 and 0.786, respectively. All of the above numbers are very close to the results of the top CASP13 groups, and it is noticeable that for overall targets, our method achieves the highest average TM-score. Notably, the comparison between our methods and CASP13 groups is not 100% rigorous, because we used domain sequence as inputs while other groups started from the whole sequence without the knowledge of domain definition, also because we only used sequence information while other groups used structure-sourced information from templates or fragments of known structures.

Figure 3 shows the detailed comparison of the TM-scores between our method and the top CASP13 groups, on targets of various alignment depths N_{eff} (i.e., the effective number of homologous sequences in MSA) and protein sizes. Clearly, our method outperforms the others for the hard targets that have low alignment depths in the MSA (see the FM target group with the smallest N_{eff} in Figure 3d), which is impressive particularly when considering that the others used structure-sourced information (e.g., templates or fragments) more or less and we used sequence information only. Consistently, correlation analysis of our method on 42 CASP13 targets (Figure S8, Supporting Information) shows weak correlation between the prediction quality and the logarithmic values of N_{eff} overall. Particularly, for FM targets, the correlation is barely present, with Pearson correlation coefficient (PCC) = 0.14 and p-value = 0.54. This phenomenon laterally hints the importance of texture information in inter-residue distance prediction for proteins with shallow alignment depths. On the other hand, in comparison to the others, our method exhibits a performance decay with the increase of target size. This problem has been partially relieved by the specific development of the low-capacity Model L to treat large proteins, but could be better solved by training using more advanced hardware to overcome the GPU memory limitation of our current facility (TITAN RTX with 24 190 MiB memory). Nevertheless, the prediction results by our method exhibit a pattern considerably distinct from those of the other state-of-the-art methods, which implies its capability of providing complementary information in practical protein structure prediction. Moreover, it is noticeable that our GAN system pipelined with CNS suite could be deployed on personal computers (PCs) with GPU cards with acceptable running time (Figure S9, Supporting Information), while the others required heavy computational resources.

Since the pipeline of A7D is already partially open-sourced, we evaluated the distance predictions of our method and A7D against the ground truths for all available CASP13 targets on the domain level. To guarantee the rigor of comparison, the same domain sequences were fed into both models, whereas the modes of predicted distributions and the real-valued distances for all residue pairs were extracted from A7D model and our GAN model, respectively, for data analysis. As shown in Figure S10, Supporting Information, in general, the distance predictions of the two methods correlate nearly equally well with the ground truths in the range of 4–16 Å. Notably,

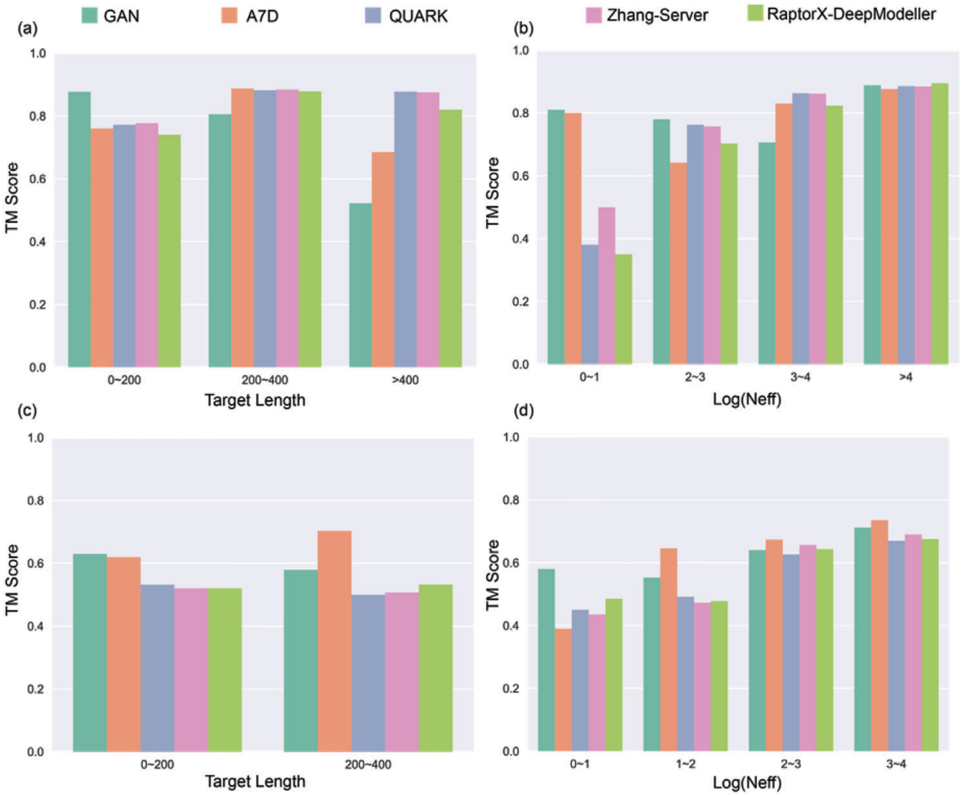

Figure 3. Comparison of TM-scores for targets by our method versus the top CASP13 groups. The TBM targets are classified by a) their length and by b) their logarithmic values of effective alignment depth (N_{eff}). The FM targets are classified by c) their length and by d) their logarithmic values of effective alignment depth (N_{eff}).

unlike the bar-screen typed pattern of A7D prediction that may arise from the discrete characteristics of multi-classification predictors, the prediction of our method exhibits a continuous pattern that more realistically reflect the nature of distance metrics.

To further check whether the good performance of our model arises from serendipity, we tested our method on the set of CAMEO hard targets as reported by Xu.[7] Three targets (2ND2A, 2ND3A, and 5B86B) were excluded from the original 41 proteins due to failure in feature generation by the third party programs (DeepCNF[19] and SPIDER3[20]). Among the remaining 38 targets (Table S10, Supporting Information), the top 1 models predicted by our method reach an average TM-score of 0.563, very close to the value (0.559) of RaptorX,[7] one of the top CAMEO servers. The robust performance of our method on CASP and CAMEO proteins supports its usefulness in practical protein structure prediction.

Structure prediction of membrane proteins is of very high value in practical usage since they are responsible for the material transport and signal transduction between cellular internal and external environments. Experimental structure determination is very hard for membrane proteins and therefore the data accumulation of known membrane protein structures is far from enough to support the regular training scheme of DNNs. However, the folding mechanism of all proteins should be the same for all proteins in the perspectives of physics and chemistry, which implies that good predictors may have good generalizability to allow the application on membrane proteins. We used 416 non-redundant membrane proteins from the PDBTM[21] set to test the generalizability of our method. Without any transfer learning, our method achieves an average TM-score of 0.602 and can fold 61.5% of the proteins (256 out of 416) into the correct topology (TM-score > 0.5). As an example, the chain A of target 5I20, an important exporter of drug/metabolite transporter superfamily in *Escherichia coli*, could be folded with very high accuracy (**Figure 4**). These results further confirm the applicability of our method on membrane proteins, although the models are trained mainly by protoplasmic soluble ones.

3. Discussion

In this work, we treated protein inter-residue distance prediction as a regression problem for the first time and precisely predicted continuous, real-valued distances merely from sequence information via an exquisitely designed GAN system. Through adversarial training procedure on the two parts of this system, that is, the generator and the discriminator, our model could learn the texture pattern of protein residue distance map. Feeding these distance predictions to the basic CNS suite produces structures with competitive model quality compared with the state-of-the-art predictors. The structure quality predicted by our system has merely weak correlation with the alignment depth in the MSA. Although trained by protoplasmic soluble proteins, the good

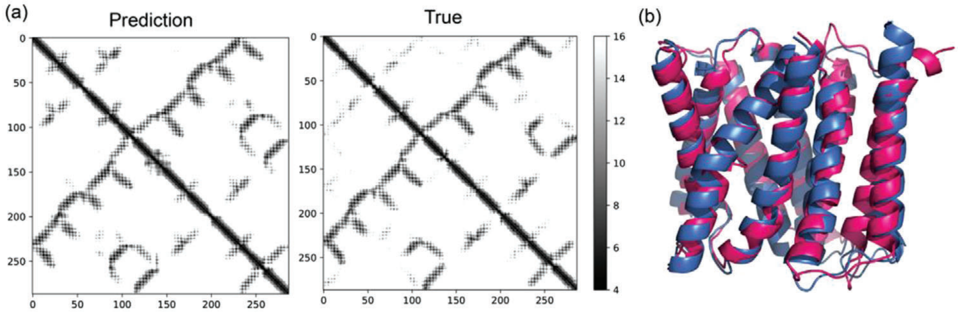

Figure 4. The prediction of our method on a membrane exporter (PDB ID: 5I20, chain A). a) Comparison of the predicted distance map by our method (left) and true distance map (right). b) Structure alignment of the best model folded using our method (red) against the crystal structure (blue).

generalizability ensures that it works for both soluble and membrane proteins. Moreover, our method could provide lightweight models that consume relatively low computation resources and can be deployed on PCs.

The prediction power of our GAN system diminishes for long proteins due to the memory limitation of our training facility. In addition, the generator of our GAN system is shallow, when compared with the 660-layer ResNet of AlphaFold. These limitations could be solved by training on more advanced hardware and/or distributed parallel training procedure in the future. Besides, lots of previous works proved the usage of metagenome databases for the MSA construction and the derivation of MSA-based input features would improve the prediction power of their models. In our next step, we would try to enhance our method further by merging metagenome databases with the regular one we are currently using.

We abandoned contact prediction because contact is a compromise when accurate distance prediction is not available. Real-valued distance has many advantages over contact, among which the most essential one is that a true real-valued distance map is a direct representation of a structure with all information included. Thus, developing differentiable distance-to-structure mapping functions that bridge our GAN system and the final structures will enable an end-to-end training procedure. Different from the dihedral angle-based end-to-end differentiable system proposed earlier,[22] which only considers local structure information of neighboring residues and fails at chirality, distance-based end-to-end differentiable system could extract global structure information for residue pairs with sequence separations of arbitrary length, and determine the chirality because only one kind of chirality should be fitted with sufficient distance restrictions. In the future, our research interest would be such an end-to-end training scheme based on GAN system presented in this paper.

We have also noticed that a new kind of GAN, called cycle GAN, which provides a generalized semi-supervised learning approach for situations of large-scale label deficiency, has been proposed recently.[23] This method may further benefit distance prediction, since many proteins have known sequences but lack structures (label for prediction). With the rapid development of high-throughput sequencing technologies and the exponential data growth of protein sequences, especially the construction of metagenome databases, it is reasonable for us to believe that the new era of "protein structure determination via sequencing" would come in the near further.

4. Experimental Section

Protein Dataset: All proteins in the training set were extracted from the SCOPe database of 2.05 version.[24] The cutoff of redundancy elimination was set as 20% sequence identity and the shortest protein of each protein family was picked out. The final training set contained 6862 proteins in total.

The methods were evaluated on four testing sets: the CASP12 set,[3a] the CASP13 set,[3b] the CAMEO set,[7] and the PDBTM set[21] of membrane proteins (choosing only one chain for each protein target). Protein name lists of all the testing sets were available at our GitHub site (see Data Availability). Considering that proteins in the training set were all determined before the CASP12 and CASP13 competitions as well as the release of the CAMEO set and that members in the PDBTM set were nonredundant to the training set, benchmarks on these testing sets could provide fair evaluation for the method.

Feature Generation: The input features of the GAN system consisted of 0D, 1D, and 2D ones. First of all, the MSAs were built through HHblits[25] from the UniProt20 database.[26] The protein length and the alignment depth constituted the 0D features. The results of DeepCNF[19] and SPIDER3[20] together with the one-hot identities and appearing frequencies of amino acids at corresponding site in the MSAs constituted the 1D features. Co-evolution information extracted from the MSAs by CCMpred[27] and mutual information, together with relative position of every site to other sites and the amount of gaps in the MSAs for every site, constituted the 2D features. The 0D features and 1D features were broadcasted to match the shape of 2D features. Notably, the 1D features were broadcasted twice, in the horizontal and vertical directions, which doubled the feature amounts. Finally, the 0D features (2 channels), 1D features (124 channels), and 2D features (4 channels) were concatenated as the input (130 channels in total).

Mapping Function: To allow the real-valued distance regression via DNN with BN layers, two different mapping functions for input features and labels, respectively, were designed. These mapping functions could map them into the interval of [−1, 1]. For simplicity, the space of the true values of features/labels was called as "real space" (RS) and the space of the mapped values as "training space" (TS), which was maintained only for training.

For every channel of the input features from RS to TS, its maximum value and minimum value were first calculated, and then its elements were mapped uniformly via linear transformation:

$$V_{\mathrm{TS}} = \frac{2V_{\mathrm{RS}} - \mathrm{Max}_{\mathrm{RS}} - \mathrm{Min}_{\mathrm{RS}}}{\mathrm{Max}_{\mathrm{RS}} - \mathrm{Min}_{\mathrm{RS}}} \tag{7}$$

where V, Max, and Min denote current value, maximum and minimum of this channel, respectively, and the subscripts represent the corresponding spaces.

The attention for label (i.e., ground-truth distance) transformation was mainly on the interval of 4–16 Å, since distances within this interval are the most valuable ones for protein folding. To disperse values within this interval to the utmost and to take all values into consideration instead of setting a cutoff and throwing some away, tanh was chosen as the mapping function here. Before using tanh, a linear transformation was conducted, which could map from the interval of [4.0, 16.0] to that of [−2.5, 2.5], where tanh had large first derivatives. The label mapping function is

$$V_{\mathrm{TS}} = \tanh\left(\frac{5 \times V_{\mathrm{RS}} - 50}{12}\right) \tag{8}$$

Since it is an invertible function, its inverse could be used to derive the final distance prediction from the outputs of DNN.

Attention Module: This module was implemented in two steps, with the first focusing on the attention of channels, and after its multiplication with the raw inputs, the second one focusing on the spatial attention of pixels. The final inputs after the process of the attention module would be:

$$\mathrm{Inputs} = (\mathrm{Raw} \times \mathrm{CAF}(\mathrm{Raw})) \times \mathrm{PAF}(\mathrm{Raw} \times \mathrm{CAF}(\mathrm{Raw})) \tag{9}$$

where CAF and PAF represent channel-wised attention function and pixel-wised attention function, respectively. In the first step (CAF), a 3-layer perceptron with a bottleneck architecture (i.e., 130–75–130) was used to process the summation of the global average and max pooling results for individual input channels. This architecture was not only light-weighted but also effective for information integration. Imaging to mix a bottle of half juice and half water, the most effective way was to squeeze the bottle neck and then loose it. ReLU activation was used in all layers except the last one, which used Sigmoid activation to force channel weights to fall in the interval of [0, 1]. In the second step (PAF), one single 7 × 7 convolution filter was used with stride 1 to scan the concatenation of the global average and max pooling results of individual input pixels and also the Sigmoid function to output pixel weights. The convolution kernel would determine whether the current pixel is an important one or not according to its neighboring zone.

MD Simulations: For all proteins in the training set, MD simulations were conducted in a water box with periodic boundary conditions applied. The boundary of the water box was set as extended by 10 Å from the edge of the protein, and the volume of the simulated system was about 354 141.7 $Å^3$ on average. To simulate the physiological environment, 160 mmol L^{-1} NaCl was added into the system (≈37 Na^+ and Cl^- ions in the water box). The specific amounts of Na^+ and Cl^- ions were set slightly different for each individual protein to ensure the electric neutrality of the system. The simulation procedure contained 3 stages: energy minimization, system heating, and equilibrium simulation. In the energy minimization stage, the low-MODe (LMOD) method was employed for 5000 steps, in which the steepest descent was applied for the first 2500 steps and then switched to the conjugate gradient descent from the next 2500 steps. In the heating stage, the volume of the system was fixed and a total of 8000 time steps were conducted with the step size of 2 femtoseconds, during which the temperature of the system was gradually heated up to 300 K from 0 K in the first 6000 time steps and was maintained at 300 K for the following 2000 time steps. In the final equilibrium simulation stage, canonical ensemble (NVT) was adopted and the simulation was run for 2 500 000 time steps, that is, 5 nanoseconds, in total. Bond interactions involving H-atoms were fixed in the last 2 stages. One structure was saved every 5000 time steps from the simulation trajectory.

Statistical Analysis: Data are presented as mean ± SD when necessary. The statistic N_{eff} is used to count the effective alignment depth for MSA, specifically in the format,

$$N_{\mathrm{eff}} = \sum_{i=1}^{N} \frac{1}{S_i} \tag{10}$$

where for N is the overall number of sequences in an MSA, i is the index of the sequence iterated in this MSA, and S_i is the count of all sequences that share ≥ 75% identity with the target sequence of index i. In Pearson correlation analysis, two-tailed chi-square test was conducted. Statistical analyses were performed using Python libraries Numpy, Scipy, and Pandas.

Code Availability: All source codes of this work are openly accessible at the GitHub site of https://github.com/Wenze-Codebase/DistancePrediction-Protein-GAN.git.

Data Availability: Protein name lists of all our testing sets are also available at our GitHub site: https://github.com/Wenze-Codebase/DistancePrediction-Protein-GAN.git.

Online Server: A user-friendly web-server is available for prediction at the site of http://structpred.life.tsinghua.edu.cn/continental.html.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (#31670723, #91746119 & #81861138009) and by the Beijing Advanced Innovation Center for Structural Biology.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

W.D. and H.G. proposed the initial idea and designed the methodology. W.D. implemented the concept and processed the results. W.D. and H.G. wrote the manuscript. Both authors read and approved the final manuscript.

Keywords

deep learning, generative adversarial networks, protein inter-residue distance, protein structure prediction, real-valued distance prediction

Received: April 8, 2020
Revised: June 6, 2020
Published online: August 10, 2020

[1] K. A. Dill, J. L. MacCallum, *Science* **2012**, *338*, 1042.

[2] a) A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler, I. W. Davis, S. Cooper, A. Treuille, D. J. Mandell, F. Richter, Y.-E. A. Ban, S. J. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popovic, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker, P. Bradley, in *Methods in Enzymology*, Vol. 487: Computer Methods, Part C, Academic Press, San Diego, CA **2011**, pp. 545–574; b) J. Yang, R. Yan, A. Roy, D. Xu, J. Poisson, Y. Zhang, *Nat. Methods* **2015**, *12*, 7; c) A. Jothi, *Protein Pept. Lett.* **2012**, *19*, 1194.

[3] a) J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, A. Tramontano, *Proteins-Struct., Funct., Bioinf.* **2016**, *84*, 4; b) J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, A. Tramontano, *Proteins-Struct., Funct., Bioinf.* **2018**, *86*, 7; c) J. Schaarschmidt, B. Monastyrskyy, A. Kryshtafovych, A. M. J. J. Bonvin, *Proteins-Struct., Funct., Bioinf.* **2018**, *86*, 51.

[4] B. Adhikari, D. Bhattacharya, R. Z. Cao, J. L. Cheng, *Proteins-Struct., Funct., Bioinf.* **2015**, *83*, 1436.

[5] a) C. Zhang, S. M. Mortuza, B. He, Y. Wang, Y. Zhang, *Proteins-Struct., Funct., Bioinf.* **2018**, *86*, 136; b) S. Ovchinnikov, D. E. Kim, R. Y.-R. Wang, Y. Liu, F. DiMaio, D. Baker, *Proteins-Struct., Funct., Bioinf.* **2016**, *84*, 67.

[6] J. Soeding, *Science* **2017**, *355*, 248.

[7] J. Xu, *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 16856.

[8] W. Ding, W. Mao, D. Shao, W. Zhang, H. Gong, *Comput. Struct. Biotechnol. J* **2018**, *16*, 503.

[9] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Zidek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, *Proteins-Struct., Funct., Bioinf.* **2019**, *87*, 1141.

[10] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, *Nature* **2020**, *577*, 706.

[11] a) I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, presented at Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, Canada, December **2014**; b) C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, in *2017 30th IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI **2017**, pp. 4681–4690 c) S. Nowozin, B. Cseke, R. Tomioka, presented at Advances in Neural Information Processing Systems 29 (NIPS 2016), Barcelona, Spain, December **2016**; d) J.-Y. Zhu, P. Kraehenbuehl, E. Shechtman, A. A. Efros, in *Computer Vision – ECCV 2016* (Eds: B. Leibe, J. Matas, N. Sebe, M. Welling), Springer International Publishing, Cham **2016**, p. 597; e) P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, in *2017 30th IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI **2017**, pp. 1125–1134

[12] a) N. Anand, P.-S. Huang, presented at ICLR 2018 Workshop, Vancouver, Canada, April–May, **2018**; b) L. Lan, L. You, Z. Zhang, Z. Fan, W. Zhao, N. Zeng, Y. Chen, X. Zhou, *Front. Public Health* **2020**, *8*, 164.

[13] a) A. T. Brunger, *Nat. Protoc.* **2007**, *2*, 2728; b) A. T. Brunger, P. D. Adams, G. M. Clore, W. L. DeLano, P. Gros, R. W. Grosse-Kunstleve, J. S. Jiang, J. Kuszewski, M. Nilges, N. S. Pannu, R. J. Read, L. M. Rice, T. Simonson, G. L. Warren, *Acta Crystallogr. Sect. D- Biol. Crystallogr.* **1998**, *54*, 905.

[14] Y. Li, C. Zhang, E. W. Bell, D. J. Yu, Y. Zhang, *Proteins* **2019**, *87*, 1082.

[15] W. Zheng, Y. Li, C. Zhang, R. Pearce, S. M. Mortuza, Y. Zhang, *Proteins* **2019**, *87*, 1149.

[16] K. He, X. Zhang, S. Ren, J. Sun, *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904.

[17] M. Tan, Q. V. Le, *2019 36th International Conference on Machine Learning (ICML)*, PMLR, Long Beach, California **2019**, *10*, pp. 6105–6114

[18] A. Boulch, *Pattern Recognit. Lett.* **2018**, *103*, 53.

[19] S. Wang, S. Weng, J. Ma, Q. Tang, *Int. J. Mol. Sci.* **2015**, *16*, 17315.

[20] R. Heffernan, Y. Yang, K. Paliwal, Y. Zhou, *Bioinformatics* **2017**, *33*, 2842.

[21] D. Kozma, I. Simon, G. E. Tusnady, *Nucleic Acids Res.* **2013**, *41*, D524.

[22] M. AlQuraishi, *Cell Syst.* **2019**, *8*, 292.

[23] J. Y. Zhu, P. Krähenbühl, E. Shechtman, A. A. Efros, in *Generative Visual Manipulation on the Natural Image Manifold* (Eds: B. Leibe, J. Matas, N. Sebe, M. Welling), Computer Vision ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9909, Springer, Cham **2016**, pp. 597–613.

[24] N. K. Fox, S. E. Brenner, J.-M. Chandonia, *Nucleic Acids Res.* **2014**, *42*, D304.

[25] M. Remmert, A. Biegert, A. Hauser, J. Soeding, *Nat. Methods* **2012**, *9*, 173.

[26] A. Bateman, M. J. Martin, C. O'Donovan, M. Magrane, E. Alpi, R. Antunes, B. Bely, M. Bingley, C. Bonilla, R. Britto, B. Bursteinas, H. Bye-A-Jee, A. Cowley, A. Da Silva, M. De Giorgi, T. Dogan, F. Fazzini, L. G. Castro, L. Figueira, P. Garmiri, G. Georghiou, D. Gonzalez, E. Hatton-Ellis, W. Li, W. Liu, R. Lopez, J. Luo, Y. Lussi, A. MacDougall, A. Nightingale, et al., *Nucleic Acids Res.* **2017**, *45*, D158.

[27] S. Seemayer, M. Gruber, J. Soeding, *Bioinformatics* **2014**, *30*, 3128.