GDFold2: a fast and parallelizable protein folding environment with freely defined objective functions

Tianyu Mi[1,2] and Haipeng Gong[1,2,*]

[1]MOE Key Laboratory of Bioinformatics, School of Life Sciences, Tsinghua University, Beijing, 100084, China

[2]Beijing Frontier Research Center for Biological Structure, Tsinghua University, Beijing, 100084, China

[*]To whom correspondence should be addressed: hgong@tsinghua.edu.cn

ABSTRACT

An important step of mainstream protein structure prediction is to model the 3D protein structure based on the predicted 2D inter-residue geometric information. This folding step has been integrated into a unified neural network to allow end-to-end training in state-of-the-art methods like AlphaFold2, but is separately implemented using the Rosetta folding environment in some traditional methods like trRosetta. Despite the inferiority in prediction accuracy, the traditional approach allows sampling various protein conformations compatible with the predicted geometric constraints, partially capturing the dynamical information. Here, we propose GDFold2, a novel protein folding environment, to address the limitations of Rosetta. On the one hand, GDFold2 is highly computationally efficient, capable of accomplishing multiple folding processes in parallel within the time scale of minutes for generic proteins. On the other hand, GDFold2 supports freely defined objective functions in order to fulfill diversified optimization requirement. Moreover, we propose a quality assessment (QA) model to provide reliable prediction on the quality of protein structures folded by GDFold2, thus substantially simplifying the selection of structural models.

Keywords protein structure prediction · structural modeling · parallelized folding · structure quality assessment

Introduction

Proteins are critical components of living organisms and provide the essential material basis for life processes. The function of a protein is typically dictated by its three-dimensional structure, as demonstrated by the general regulation mechanism of protein function and property through conformational change[1]. Hence, acquisition of accurate three-dimensional structures of proteins is not only indispensable for the mechanistic exploration in life sciences but also pivotal in advancing fields such as medicine and pharmacology. Unlike traditional protein structure determination that relies on costly and time-consuming biological experiments, advancement of deep learning technologies has elicited unprecedented progress in the field of protein structure prediction, allowing accurate and fast inference of protein structures from their amino acid sequences, as witnessed by the critical assessment of protein structure prediction (CASP) competition[2] in the past decade. In CASP12, RaptorX-Contact[3] first applied the deep residual convolutional neural network to improve the prediction of inter-residue contacts from multiple sequence alignment (MSA), which effectively constrained the conformational sampling space in the downstream structural modeling process. AlphaFold1[4] upgraded the contact prediction to distance prediction in CASP13, followed by trRosetta[5] that proposed to predict the inter-residue orientations as additional geometric constraints. Distributions of these predicted geometric constraints could be converted to pseudo energies in protein folding environments like Rosetta[6] to construct the compatible protein structural models through gradient-based optimization. AlphaFold2[7] in CASP14, followed by RoseTTAFold[8], integrated the inter-residue geometry prediction and geometry-constrained structural modeling in a unified neural network, which markedly suppressed the prediction errors through end-to-end training, enhancing the precision of monomeric protein structure prediction to a level comparable to experimental resorts. Subsequent efforts included but not limited to open-sourcing the AlphaFold2 framework (OpenFold[9] and UniFold[10]), prediction of protein-involved complex structures (AlphaFold-Multimer[11], DeepMSA2[12], RoseTTAFold All-Atom[13] and NeuralLPexer[14]), replacing the MSA processing by the protein language model to allow structure prediction from single sequences (ESMFold[15], OmegaFold[16], HelixFold-single[17], RGN2[18] and trRosettaX-Single[19]), as well as exploring multiple conformational states through MSA manipulation in AlphaFold2 (AF-Cluster[20]).

Despite the great success, AlphaFold2 has limited power in exploring dynamic protein structures. AF-Cluster partially overcomes this problem by iteratively feeding MSA sub-clusters to AlphaFold2, but still confronts the challenges of the arbitrariness and subjectiveness in MSA manipulation, randomness and uncertainty in the prediction results, as well as the high demand for computational resources. As a contrast, the Rosetta-based folding procedure of trRosetta and RoseTTAFold are, in principle, capable of sampling various conformations compatible with the predicted geometric constraints and thus capture the protein dynamics to some extent, despite the mildly declined prediction accuracy due to the isolation of constraint prediction and structural modeling. The Rosetta environment employed by the trRosetta/RoseTTAFold pipeline, however, has the follow deficits: 1) the folding process is slow, frequently taking hours for a medium-sized polypeptide chain; 2) the folding framework is rigid in that all predicted geometric constraints have to be artificially converted into the style of statistical potentials. The second point, particularly, restricts its application in modeling protein structures using diversified constraints. For instance, in our recent work, we developed a lightweight protein structure prediction method SPIRED[21], which utilizes 2D information to predict L (*i.e.* the number of residues) sets of C_α atom coordinates, each taking the local coordinate system of one individual residue as reference. Modeling the protein structures compatible with such prediction results is feasible in Rosetta only by converting the information-abundant C_α coordinates into distributions of inter-residue distances, a procedure that clearly leads to severe information loss. In another work, we developed a novel protein structure prediction framework Cerebra[22], which significantly improves the computational efficiency by predicting multiple sets of atomic coordinates in parallel and leveraging their complementarity for prediction error suppression. In each set of the prediction results, Cerebra outputs the relative translations and rotations (in quaternions) of the local coordinate systems of all

residues in the reference frame of one anchor residue. Unfortunately, structural modeling given such prediction results is unsupported by the available energy definitions of the Rosetta environment. Therefore, the development of a more efficient protein folding environment with higher degree of freedom in the objective function definition is still awaiting investigation.

In this work, we developed a novel protein folding environment, GDFold2 (gradient descent based folding environment version 2, distinguished from GDFold[23]), to fulfill the aforementioned demand. Firstly, it is capable of integrating various sources of predicted geometric information with statistical constraints to rapidly construct satisfactory all-atom protein structures. Secondly, it supports highly flexible and fully customizable constraining loss functions, which can be rewritten by users according to their own definitions and requirements. Here, we provide three predefined folding modes as references for users, which aim for generating all-atom protein structures compatible with the prediction results of SPIRED, Cerebra and trRosetta/RoseTTAFold, respectively. Moreover, GDFold2 is highly computationally efficient, faster than pyRosetta by at least one order of magnitude, and allows parallelized structural modeling. We also trained a quality assessment (QA) model to predict the ranking of quality for the structural decoys generated by multiple GDFold2 folding events. This scoring model can reliably assist users in selecting high-quality structural models that are compatible with their predicted constraints.

Methods

As shown in Figure 1, the overall process of our method consists of two main stages. In the first stage, the geometric information predicted from various models is converted into constraints and protein structures are folded by GDFold2 based on these constraints. In the second stage, the multiple structural decoys produced by GDFold2 are evaluated by the QA model for quality ranking and model selection.

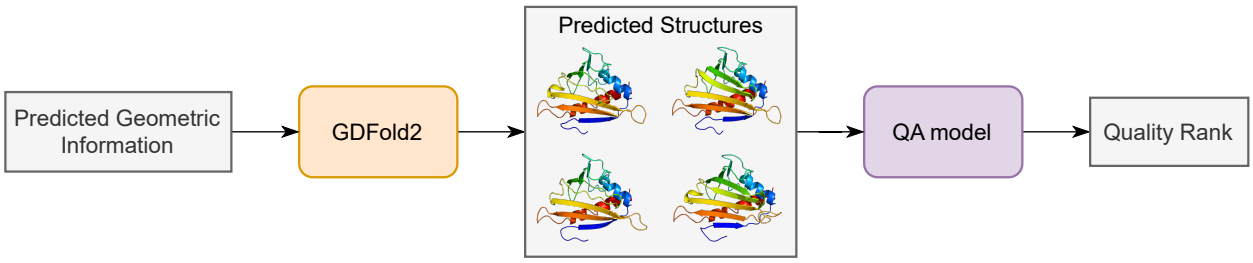


Figure 1. Overview of the entire method.

Overview of the GDFold2 pipeline

The pipeline of GDFold2 is shown in Figure 2. In this folding environment, 2D geometric information predicted from various methods and statistics summarized from high-resolution protein structures are converted into geometric constraints (Figure 2A) and statistical constraints (Figure 2B), respectively, which are then taken as optimization aims for the user-defined loss functions in the neural network (named atomic coordinate network) for atomic coordinate generation. Specifically, coordinates of C_α atoms and relative rotations of the local coordinate systems of all residues are registered as learnable parameters of the atomic coordinate network, whose values are automatically optimized during the gradient-descent-based minimization of loss functions (Figure 2C). Upon convergence, coordinates of all atoms could be reconstructed from the optimized learnable parameters, based on prior knowledge of the intra-residue and peptide-bond internal coordinate information (Figure 2D).

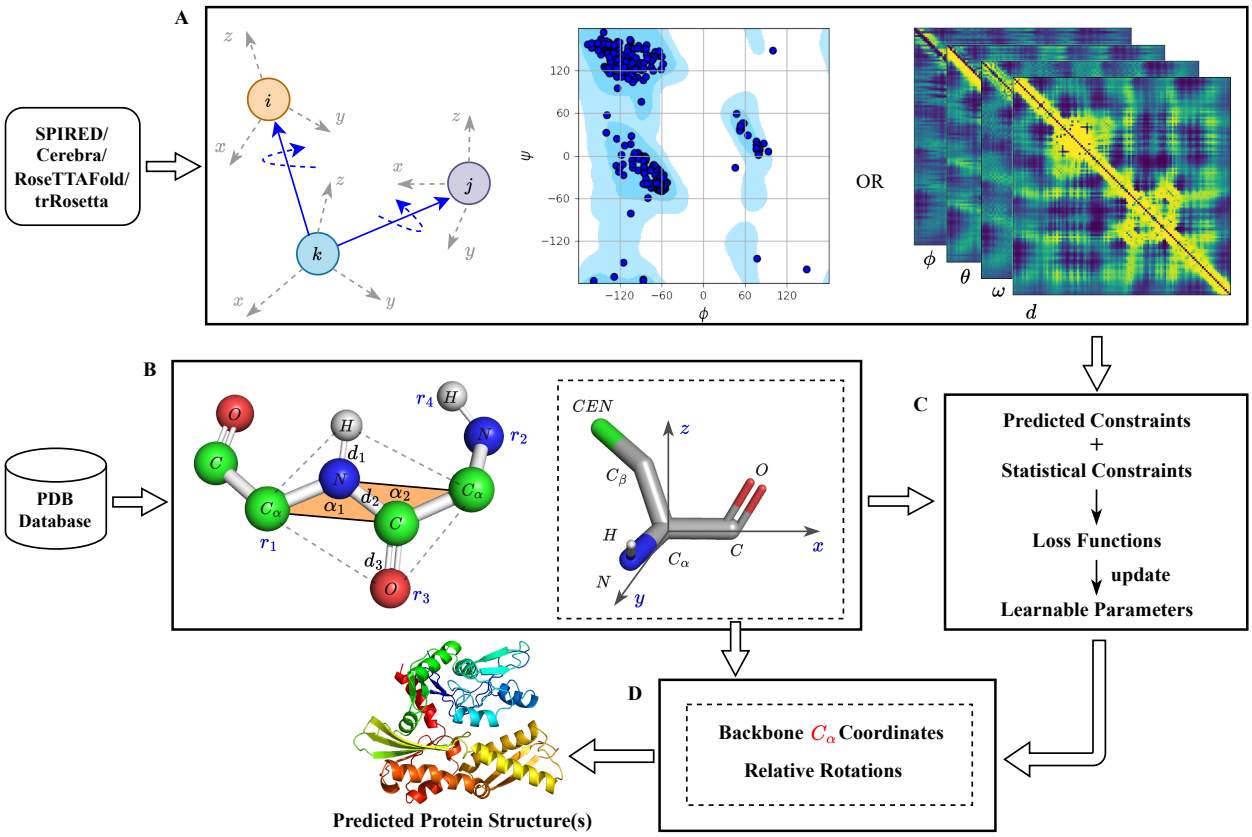


Figure 2. Schematic representation of the GDFold2 pipeline. A) Various types of predicted geometric information are obtained from different models. The relative translations and/or rotations of residues, backbone dihedral angles, probability distributions of inter-residue distances and orientational angles are shown as examples here, from left to right. **B)** Statistical parameters describing the general characteristics of polypeptide chains are obtained from high-resolution protein structures. **C)** In the atomic coordinate network, the C_α coordinates and relative rotations of all residues are learnable parameters, whose values are optimized during the minimization of the loss functions. **D)** The all-atom protein structure could be reconstructed from convergent parameters of the atomic coordinate network and statistical information of chemical bonds in polypeptides (shown in panel **B**).

Geometric constraints

Different protein structure prediction methods often output diversified geometric information as prediction results. In SPIRED and Cerebra, a set of K residues are selected as anchor residues and their local coordinate systems are established with the C_α atoms placed at origins (Figure 2B). Prediction results include translational vectors of all residue within each reference frame of the K anchor residues (*e.g.*, blue vectors departing from residue k in Figure 2A) as well as main-chain dihedral angles. Cerebra predictions also include the residue rotations (in quaternions), which in combination with translations allow estimation of the positional vectors (**v**) of C, N and C_β atoms in addition to the C_α atoms within each of the K anchor reference frames. In the folding process of GDFold2, the same relative vectors could be computed from the learnable parameters, which thus allows evaluating the degree of consistency between protein structure under optimization and the prediction results, as shown in Equation 1:

$$\mathcal{L}_{vec} = \begin{cases} \langle \| \mathbf{v}_{struct} - \mathbf{v}_{label} \|_2 \rangle & \text{mode = Cerebra} \\ \\ \langle \| \mathbf{C}_{\alpha,struct} - \mathbf{C}_{\alpha,label} \|_2 \rangle & \text{mode = SPIRED} \end{cases}, \tag{1}$$

where $\langle \| \cdot \|_2 \rangle$ denotes the L_2 norms averaged over all residues and over all reference frames, while $struct$ and $label$ refer to the protein structure under optimization and the label converted from the predict results of SPIRED/Cerebra, respectively. The detailed implementation of $\mathcal{L}_{vec}$ could be found in Algorithm 1 and Algorithm 2 for Cerebra and SPIRED, respectively. Moreover, main-chain dihedral angles estimated from the protein structure under optimization are also evaluated against prediction results in a similar manner (see Algorithms 3, 4 and 5 for details).

As for trRosetta/RoseTTAFold, the predicted probability distributions of inter-residue distances and orientational angles are converted into loss functions in the format of statistical potentials, following the "trFold.py" script of RoseTTAFold.

Statistical constraints

Proteins should follow the chemical requirement of polypeptides during conformational changes. Hence, the universal characteristics of polypeptides summarized from high-resolution experimental structures could serve as statistical constraints. We analyzed protein structures with resolution < 3 Å in the CATH[24] database and extracted the following statistical parameters: peptide bond length, distance between neighboring residues, angles between the adjacent covalent bonds within the peptide plane, van der Waals radii of backbone atoms, vectors of carbonyl oxygen (O) and hydrogen (H) atoms in the peptide plane coordinate system and residue coordinate systems of common amino acids (Figure 2B, see **Supplementary Materials** for details). Incidentally, to reduce the computational complexity, we substitute the side-chain of each amino acid residue with a centroid atom that has a larger van der Waals radius (see *CEN* in Figure 2B). Constraints constructed based on the above statistical data are applicable to all monomeric protein structure modeling. Detailed implementation about the construction of statistical constraints could be found in Algorithms 6, 7 and 8.

Atomic coordinate network

Generally, protein folding refers to the physical process in which a protein transits from a random coil to a specific functional 3D structure. GDFold2 translates this process into the optimization of atomic coordinates of all amino acid residues based on predicted geometric information and statistical constraints. To achieve this goal, we establish a network with the C_α coordinates (in vectors) and the relative rotation matrices (in Euler angles) of all residues as learnable parameters, which are initialized randomly at the beginning. As mentioned previously, all constraining items are constructed in the form of loss functions required for updating neural network parameters, leveraging the reverse automatic differentiation mechanism of PyTorch[25], which significantly enhances the computation speed. Convergence of the overall loss indicates that the current atomic coordinates obtained from the network conform to the predicted geometric features and statistical patterns of polypeptides. Moreover, the batch size of learnable parameters existing in the form of high-dimensional tensors corresponds to the number of predicted structures, allowing the network to parallelly update multiple sets of atomic coordinates, thereby greatly improving the prediction efficiency.

3D protein structure modeling

The three-dimensional protein structure is built in the global Cartesian coordinate system. The local coordinate system of each amino acid residue is defined in the same way as AlphaFold2[7], by placing the C_α atom at the origin, C atom on the x-axis and N atom in the xy-plane. Thus, given the optimized C_α coordinates and relative rotation matrices of all residues, the positional vectors of all atoms could be inferred in the global frame, based on prior knowledge about the intra-residue and peptide-bond internal coordinate information (see **Supplementary Materials**). The detailed implementation could be found in Algorithms 9, 10, 11, 12 and 13.

The full folding process of GDFold2 is shown in Algorithm 14. The structural models output by GDFold2 are further optimized by a modified pyRosetta-based FastRelax procedure, with the starting backbone coordinates applied as positional restraints (in terms of quadratic potentials). This relax process enables sampling of the complete side-chain (instead of the centroid atoms) as well as further improvement over the backbone.

Overview of the quality assessment pipeline

As shown in Figure 3, the QA model adopts the graph neural network (GNN) architecture, which reads in node and edge features collected from an input protein structure and outputs the evaluation score for the structural quality. Details about the QA model are described in the following sections in four aspects: data preparation, feature design, network architecture and model training.

Data preparation

We randomly selected 4,169 and 5,452 target proteins from the training sets of SPIRED and Cerebra, respectively, and performed model inference to obtain the predicted geometric information. These targets range from 50 to 450 residues in length and have an average plDDT of 0.8735. To obtain decoy structures of varying predictive quality, we first designed the mask setting according to Equation 2:

$$M_i = \begin{cases} \text{pLDDT}_i & n = 1 \\ m_i = \begin{cases} 0 & R_i < \frac{r \times (n-2)}{N-2} \\ 1 & R_i > \frac{r \times (n-2)}{N-2} \end{cases} & n < 1 \leq N \end{cases}, \tag{2}$$

where M_i represents the masking weight for the ith residue, n represents the nth decoy structure, r and N are the upper limit of mask ratio and the number of decoy structures to be generated for each target, respectively, while R is a randomly generated float number between 0 and 1. Due to the distinct sensitivity to the masking weight, the setting of r is different according to the source of predicted geometric information (0.8 for Cerebra and 0.4 for SPIRED). N is determined by the number of anchor residues specified by Cerebra and SPIRED. These masks, acting as tensor weights, were then multiplied to the tensors of predicted constraints in the same shape of $N \times L$ to tune the scales of individual items. Hence, this procedure masked the predicted geometric information from loss calculation for randomly selected residues. Next, we utilized GDFold2 to parallelly fold proteins based on the masked geometric information. Finally, we obtained a total of 259,875 decoy structures for the training of QA model.

Feature design

A protein structure is represented by a graph composed of nodes and edges in the QA model (Figure 3B). The node features characterize the information for each residue. Firstly, each residue of the protein is transformed into a 20

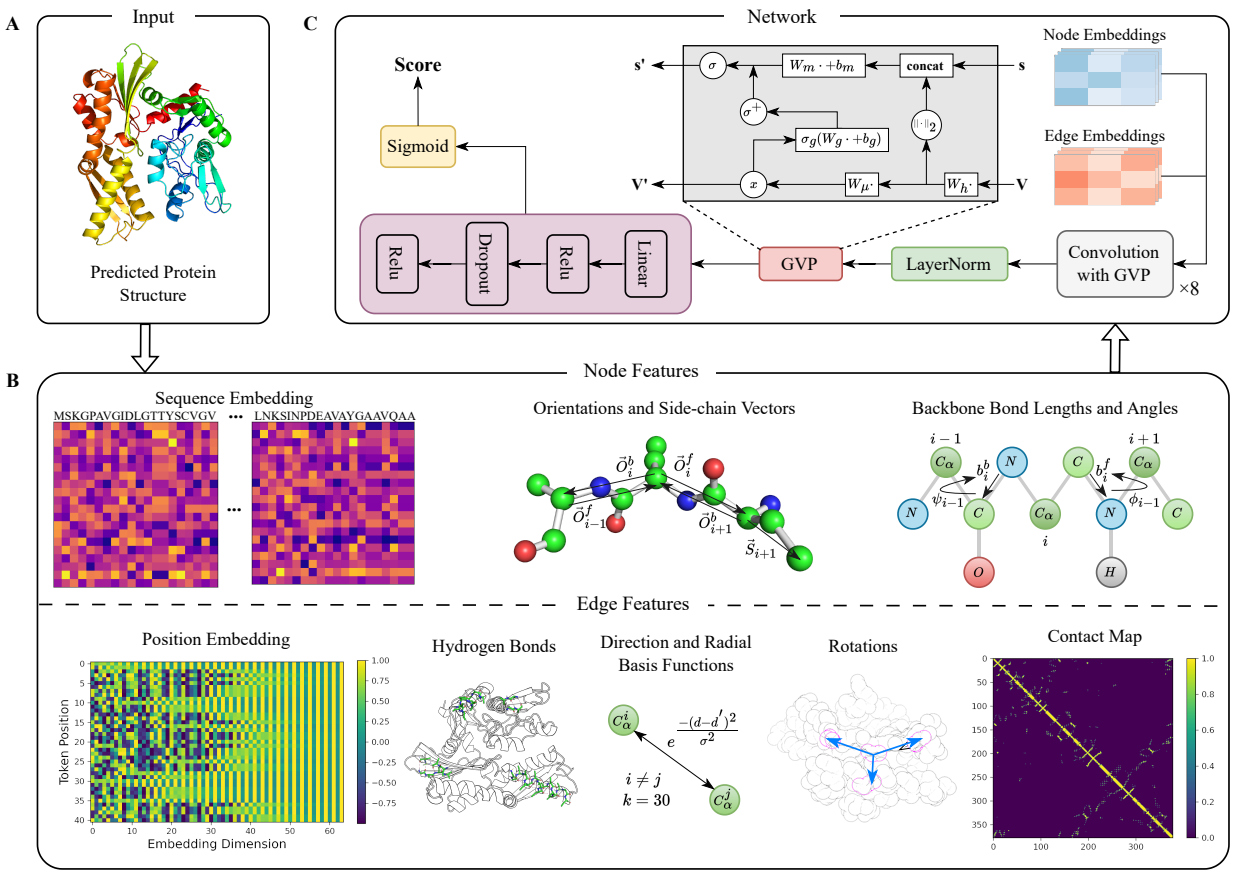


Figure 3. Schematic representation of the quality assessment pipeline. A) The protein structure under evaluation is taken as the raw input. **B)** Node and edge features are extracted from the input protein structure. Node features include sequence embedding, main-chain orientational and side-chain directional vectors as well as backbone bond lengths and angles. Edge features include positional embedding, one-hot encoding of hydrogen bonds, direction and distance between each pair of C_α atoms, C_β contact map as well as relative rotations between each residue and its top K spatially nearest neighboring residues. **C)** Network architecture of the QA model. The network embeds the entire features (in panel **B**) into node and edge embeddings, and utilizes graph-based message passing mechanism to update and transfer the message between layers. A sigmoid activation function is connected with the last layer to output the evaluation score, which falls between 0 and 1.

dimensional tensor to represent the sequence information. Then, in order to extract the topology features between residue pairs and characterize the overall conformation, we utilize the forward ($\mathbf{O}_i^f$) and backward orientations ($\mathbf{O}_i^b$) to represent the relationship between adjacent residues, and use a vector ($\mathbf{S}_i$) to describe the direction of side-chain centroid. Lastly, backbone bonds and angles are represented as scalar features. Specifically, peptide bond lengths are also embedded into forward (b_i^f) and backward (b_i^b) values to characterize the distance features between residue r_i and its succeeding residue r_{i+1} and preceding residue r_{i-1}, respectively. Additionally, backbone dihedral angles ϕ, ψ and ω are embedded into tensors of $(L-1) \times 2$ dimension through sine and cosine triangular functions, respectively, to describe the relationship between adjacent planes.

The edge features in graph are used to characterize the information between residue pairs. Here, we apply five edge features. Firstly, for each residue i, the top 30 spatially nearest neighboring residues are obtained by calculating the distance between their C_α atoms. The unit vector from residue i to its neighboring residue j is the direction feature, calculated as $(\mathbf{C}_\alpha^j - \mathbf{C}_\alpha^i)/\|\mathbf{C}_\alpha^j - \mathbf{C}_\alpha^i\|_2$. Then, we utilize the sinusoidal function and Gaussian radial basis function

(RBF) to encode the direction and distance, respectively, to represent the relationship between these residue pairs. Embedding dimensions in direction and σ in Gaussian RBF are hyper-parameters. In addition, hydrogen bonding is a kind of intra-molecular interaction that significantly contributes to the formation of protein secondary structures. Thus, we use the DSSP[26] vacuum electrostatics model to encode the hydrogen bonds. Next, the relative rotations between each residue and its top 30 spatially nearest neighboring residues are calculated and then converted into the quaternion form to characterize the orientational relationship between residue pairs. Lastly, we use the contact map of C_β atoms with the contact cutoff of 8 Å to represent the overall structure information.

Network architecture

The neural network takes as input the protein graph designed above and utilizes the message passing mechanism in Equation 3 and Equation 4: messages from neighboring nodes and edges are used to update node embeddings at each graph propagation step.

$$\mathbf{h}_m^{(j\to i)} := g\left(\text{concat}\left(\mathbf{h}_v^{(j)}, \mathbf{h}_e^{(j\to i)}\right)\right) \tag{3}$$

$$\mathbf{h}_v^{(i)} \leftarrow \text{LayerNorm}\left(\mathbf{h}_v^{(i)} + \frac{1}{k'}\text{Dropout}\left(\sum_{j:\mathrm{e}_{j\to i}\in\varepsilon} \mathbf{h}_m^{(j\to i)}\right)\right) \tag{4}$$

Here, g is a sequence of graph vector perception (GVP) modules. GVP module is designed to learn the vector-valued and scalar-valued functions for geometric vectors and scalars, respectively. In a GVP module, two linear projections are employed for the transformation of scalar and vector features. Before the transformation, scalar features are concatenated with the L_2 norms of vector features, enabling the extraction of rotation-invariant information from vector features (Figure 3C, see Algorithm 15 for more details). $\mathbf{h}_v^{(i)}$ and $\mathbf{h}_e^{(j\to i)}$ are the embeddings of the node i and directional edge $(j \to i)$, respectively, whereas $\mathbf{h}_m^{(j\to i)}$ represents the message passing from node j to node i. k' is equal to the number of incoming messages k unless the protein contains fewer than k amino acid residues.

Model training

We randomly split 9,621 target proteins into training and validation sets with a ratio of 8:2. All decoy structures for one target protein are organized into one training mini-batch, and each training epoch involves traversing all target proteins. To balance the evaluation of global quality and local quality of predicted structures, we take the average value of lDDT and TM-score as the evaluation score and set the mean soft Spearman correlation coefficient[27] between predicted score and true evaluation value as the training objective (see following items for more details about evaluation metrics). Model training on a TITAN RTX 2080Ti GPU took 70 hours in total.

lDDT[28] is a superposition-free score that indicates the difference in local inter-residue distances between the predicted structure and reference structure (Equation 5). In this study, we only calculate the lDDT score for C_α atoms.

$$\text{lDDT} = \frac{1}{4} \times \sum_{t\in\{0.5,1,2,4\}} \sum_{i=1}^{L} \frac{\sum_{j,|i-j|\geq r, D_{ij}\leq 15} \mathbb{I}(|D_{ij} - d_{ij}| < t)}{\sum_{j,|i-j|\geq r, D_{ij}<15} \mathbb{I}}, \tag{5}$$

where $\mathbb{I}$ is the indicator function, d and D are the inter-residue distances in the predicted structure and reference structure, respectively, and r is the threshold for sequence separation.

TM-score[29] is a metric used to assess the topological similarity between protein structures. According to Equation 6, TM-score ranges from 0 to 1, with the value of 1 indicating a perfect match between structures.

$$\text{TM-score} = \text{Max}\left[\frac{1}{L_N}\sum_{i=1}^{L_r}\frac{1}{1+(\frac{d_i}{d_0})^2}\right], \tag{6}$$

where d_i represents the distance between the ith aligned residue pairs, d_0 is a normalization scale, L_N denotes the original length of the protein, and L_r represents the number of aligned residues.

Score is calculated from the average of TM-score and lDDT and is used as the evaluation metric of the QA model (Equation 7). It is not only sensitive to the local difference, but also indicates the degree of global match.

$$\text{Score} = \frac{(\text{TM-score} + \text{lDDT})}{2} \tag{7}$$

Spearman correlation coefficient[30] (ρ) is commonly used to describe the strength of a monotonic relationship between two variables. According to Equation 8, Spearman correlation coefficient is calculated by utilizing the ranked values of both variables (X, Y).

$$\boldsymbol{\rho}_{X,Y} = \frac{Cov(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}}, \tag{8}$$

where Cov and σ denote the covariance and standard deviation, respectively, while R is the ranking operation.

Results

The folding process and performance of GDFold2

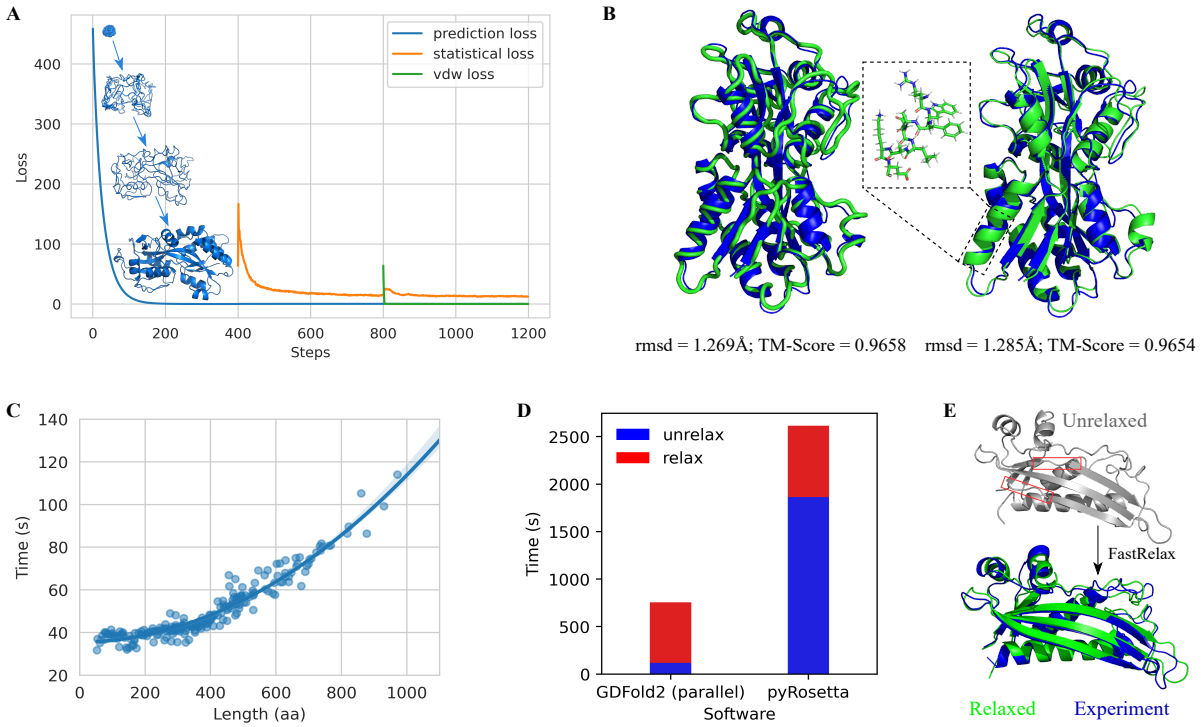


Figure 4. Folding process and performance of GDFold2 A) Evolution of loss functions and the protein structure in the folding process of GDFold2. The horizontal axis represents the number of optimization steps and the vertical axis represents the value of applied loss. At steps of 400 and 800, the statistical loss and van der Waals loss is added to the optimization process, respectively. **B)** Comparison of the raw C_α coordinates output by SPIRED (green, left) with the all-atom protein structure folded by GDFold2 (green, right), both superposed with the native structure (blue). **C)** Analysis of folding time for GDFold2. The horizontal axis represents the length of proteins and the vertical axis represents the time taken for GDFold2 folding. **D)** Comparison of running time. For the sake of fairness, GDFold2 and pyRosetta take the same geometric information (CASP14 target T1025) predicted by RoseTTAFold as input. Blue bars indicate the folding time without the pyRosetta-based FastRelax process, while the red bars represent the time of pyRosetta-based relax. **E)** A self-defined pyRosetta-based FastRelax method with the starting backbone coordinates engaged as positional restraints. It is specially designed for GDFold2-folded proteins in order to optimize the secondary structure without compromising the high prediction accuracies of SPIRED/Cerebra models.

The folding process of GDFold2 can be divided into three main stages (Figure 4A), separated by the step points at which new loss terms are applied. The first stage optimizes the backbone structure of the target protein using the loss functions designed for the predicted geometric constraints. Due to the random initialization of learnable parameters in the atomic coordinate network, all atoms are located within a small sphere at the beginning. After the imposition of the predicted geometric constraints, the protein structure expands in dimension, gradually exhibiting features of the predicted geometry. In the second stage, multiple statistical constraint terms are employed as optimization objectives to fine-tune the local conformations, aiming to rectify the inconsistency between predicted geometric information and statistical regularities of polypeptides. After the first two stages, the protein backbone structure generally meets the prediction requirements. In the third stage, the van der Waals repulsion constraint is applied to eliminate the possible clashes between atom pairs and then a modified pyRosetta-based FastRelax procedure is engaged to sample the full side-chain conformations based on the coordinates of centroid dummy atoms (*i.e. CEN*) output by GDFold2.

Besides the full sampling of side-chain rotamers, the main-chain secondary structure contents are also improved in this stage. To accelerate the folding process, the learning rate is set to a large value (0.6) for the first stage but tuned to a small value (0.02) for the second and third stages. Incidentally, whenever a new constraint term is introduced as an optimization objective, the loss value rises up abruptly but anneals down rapidly, supporting their successful constraining effects.

The raw prediction results of SPIRED contain coordinate information for C_α atoms only. However, by converting the C_α coordinates into geometric constraints, GDFold2 is capable of replenishing the missing atoms. As shown by the example in Figure 4B, the all-atom model provided by GDFold2 demonstrates abundant structural details, *e.g.*, secondary structure contents and side-chain conformations, while the nearly negligible decline in C_α TM-score and RMSD supports the compatibility of this structural model with the raw SPIRED prediction. Hence, the non-statistical-potential-style loss function in GDFold2 successfully complements the 2D geometric prediction results by statistics derived from high-resolution protein structures, nearly without causing information loss. More GDFold2 folding cases could be found in the Supplementary Materials of our SPIRED paper[21]. Unlike SPIRED, Cerebra prediction results involve full information required for all-atom structural reconstruction. However, GDFold2 folding based on Cerebra-type constraints can improve the local structural quality due to the inclusion of the statistical constraints in the optimization process (see **Supplementary Materials** for details).

We evaluated the folding time of GDFold2 on 219 randomly selected protein targets with length ranging from 50 to 1000 residues, using the SPIRED predictions as geometric constraints (maximal folding step = 600; batch size = 1). According to the evaluation curve (Figure 4C), GDFold2 accomplishes folding within 1 minute for proteins of less than 400 residues. Even for large proteins of ~1000 residues, the full folding process only takes 2 minutes, significantly faster than the traditional Rosetta-based folding. Subsequently, in order to quantitatively compare the running time of GDFold2 and pyRosetta, we took the RoseTTAFold-predicted inter-residue distance and orientation distributions of CASP target T1025 as input, and employed GDFold2 (maximual folding step = 600; batch size = 10) and pyRosetta (default parameters from RoseTTAFold) to build the structural models, respectively. At the same level of quality for the generated structural models (see **Supplementary Materials** for details), GDFold2 is markedly more efficient than pyRosetta (Figure 4D): the average folding time per structure of GDFold2 is only 2.89 % of that of pyRosetta.

In the third stage of GDFold2 folding process, the van der Waals repulsion constraint employed for clash elimination is inclined to push the side-chain centroid dummy atoms of large residues farther away, thus disrupting the potential hydrogen bonds and secondary structure contents (Figure 4E, labeled by red boxes on the gray structure). The pyRosetta-based FastRelax procedure can effectively fix this problem, but frequently elicits large deviation in the positions of backbone atoms, leading to considerable inconsistency between the relaxed protein structures and SPIRED/Cerebra predictions. To address this trouble, we implemented a specialized pyRosetta-based FastRelax procedure for GDFold2, by taking the backbone atom coordinates of GDFold2-folded structures as positional restraints during the simulated annealing processes of FastRelax. As shown in Figure 4E, the self-defined FastRelax method optimizes both the side-chain conformations and the secondary structure contents of the GDFold2-folded structure (structure in green), while maintaining the high predictive accuracy of the main-chain structure.

Performance of the QA model in ranking and selection

We employed the highly efficient GDFold2 pipeline to generate a large amount of decoy structures based on randomly masked SPIRED and Cerebra predictions for a number of proteins in their training sets (see **Methods**) and used these decoys to train the QA model. Consequently, to avoid information leaking, during the evaluation of our QA model, we chose proteins in the CAMEO testing dateset[21] that are absent in the SPIRED/Cerebra training sets, and selected

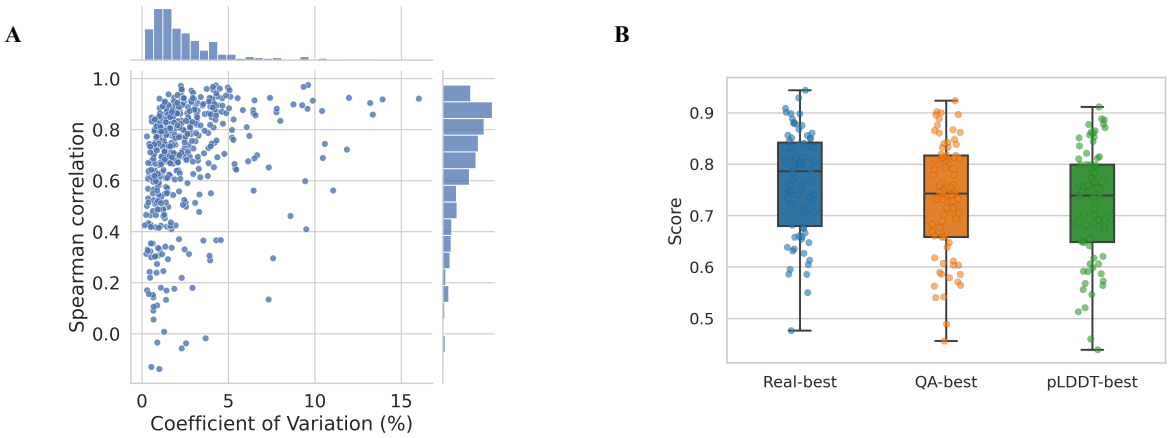


Figure 5. QA model performance in ranking and selection A) Analysis on the structure decoys produced by GDFold2 based on RoseTTAFold-type constraints. Each dot in the plot denotes a protein. The horizontal axis represents the coefficient of variation (c_v) of all GDFold2-folded protein decoys, while the vertical axis represents the Spearman correlation coefficient (ρ) between the Score predicted by the QA model and the true evaluation score. **B)** Comparison of the best structures selected by the QA model and by plDDT reported by Cerebra.

RoseTTAFold predictions as constraints to generate 24 structural models by GDFold2 for each protein target. Generally, the structural models produced for each individual protein target present certain degree of structural dissimilarity, with an average deviation (*i.e.* max - min) of 0.0714 in the evaluation score (deviation of 0.048 in TM-score and 0.1016 in lDDT), since the RoseTTAFold prediction results (*i.e.* distributions of inter-residue distances and orientational angles) can only impose weak constraining effects. Here, we use the coefficient of variation (c_v) to measure the degree of variation among the structural models generated by GDFold2:

$$c_v = 100\% \times \frac{\sigma}{\mu}, \tag{9}$$

where σ and μ are the standard deviation and the mean of the true evaluation scores of the structural models, respectively. The total of 476 CAMEO testing protein targets have an average c_v of 2.53%, with 89.9% of the targets showing c_v values of < 5% (Figure 5A), reinforcing the small variation among the generated structural models. However, the QA model shows a strong capability of discerning these slightly different structural models: the Spearman correlation coefficient between the predicted Score and true evaluation score exceeds 0.6 for 74.6% of the testing targets, attaining an average value of 0.6876 for all targets. Expectedly, our QA model is more powerful at predicting the ranking of quality for structural models with larger degree of structural variations (Figure 5A).

Table 1. Comparison of the structural model selection by the QA model and the Cerebra plDDT

Method	TM-score	lDDT	Truly Best as the Best	Truly Best in Top-3
plDDT	0.7153	0.7264	4/76	11/76
QA model	0.7269	0.7418	11/76	28/76

Unlike the trRosetta/RoseTTAFold outputs, the prediction results of SPIRED and Cerebra are composed of atomic coordinate information, which intrinsically exerts strong constraining effects during GDFold2 folding. Hence, in the absence of masking, the structural models generated from the same set of SPIRED/Cerebra predictions usually show a high degree of structural similarity (see **Supplementary Materials** for the effects of masking). In practical

structural prediction, however, multiple model checkpoints with different random seeds are frequently collected as an ensemble to further enhance the prediction accuracy. Consequently, a good QA model should be able to select high-quality structural models from the ensemble of GDFold2-folded structures using constraints predicted from various combinations of checkpoints and random seeds. Here, we evaluated our QA model in this aspect on 76 CAMEO protein targets. Specifically, we randomly selected 20 combinations of seeds and checkpoints (4 random seeds $\times$ 5 checkpoints) for Cerebra, fed the same MSA inputs to these Cerebra models to make prediction, and then employed GDFold2 to fold proteins using each individual set of prediction results as geometric constraints. Thus, for each of the 76 protein targets, we generated 20 structural models, corresponding to various combination of seeds and checkpoints. As shown in Figure 5B, the best structural models selected by the QA model have an average score of 0.7343, closer to the truly optimal structures (average score of 0.7643) than those picked up by the raw plDDT values reported by Cerebra (average score of 0.7209). Similarly, as shown in Table 1, the best structures selected by the QA model outperform those chosen based on plDDT in both the average TM-score and the average lDDT. Moreover, the QA model exhibits a stronger power at ranking the truly optimal structure as the best and as among the top 3. More detailed results could be found in the **Supplementary Materials**.

Conclusion

In this study, we have developed a novel protein folding environment, GDFold2, which allows freely defined objective functions for protein structure generation. Loss functions have been designed as examples for three application scenarios, with highly diversified geometric information predicted from various prediction methods. In all scenarios, the predefined loss function can effectively constrain the protein folding process and prompt the production of high-quality protein structural models in combination with statistical constraints derived from high-resolution protein structures. The overall framework is customizable, allowing users to define their own loss functions for fulfilling specific practical requirements. In comparison with popular protein folding environments like Rosetta, GDFold2 exhibits advantages in the significantly reduced folding time, the availability of parallelized folding and the high degree of freedom in the definition of objective functions. Additionally, a QA model has also been developed for reliably estimating the quality ranking of protein structures folded by GDFold2, which further simplifies the model selection process.

Data availability

All data are available at https://zenodo.org/doi/10.5281/zenodo.10809439.

Code availability

The codes of GDFold2 are available at https://github.com/Gonglab-THU/GDFold2.

Author contribution

H. G. proposed the concept and theory. For the GDFold2 and QA models, T. M. and H. G. proposed the initial model design, while T. M. implemented coding, model training and testing as well as data analysis. T. M. and H. G. wrote the manuscript. Both authors agreed with the final manuscript.

References

1. Buxbaum, E. in *Fundamentals of Protein Structure and Function* 15–64 (Springer International Publishing, 2015). https://doi.org/10.1007/978-3-319-19920-7_2.
2. Alexander, L. T., Durairaj, J., Kryshtafovych, A., *et al.* Protein target highlights in CASP15: Analysis of models by structure providers. *Proteins: Structure, Function, and Bioinformatics* **91,** 1571–1599. ISSN: 1097-0134. http://dx.doi.org/10.1002/prot.26545 (2023).
3. Wang, S., Sun, S., Li, Z., *et al.* Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *PLOS Computational Biology* **13** (ed Schlessinger, A.) e1005324. ISSN: 1553-7358. http://dx.doi.org/10.1371/journal.pcbi.1005324 (2017).
4. Senior, A. W., Evans, R., Jumper, J., *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577,** 706–710. https://doi.org/10.1038/s41586-019-1923-7 (2020).
5. Yang, J., Anishchenko, I., Park, H., *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences* **117,** 1496–1503. https://doi.org/10.1073/pnas.1914677117 (2020).
6. Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., *et al.* The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation* **13,** 3031–3048. https://doi.org/10.1021/acs.jctc.7b00125 (2017).
7. Jumper, J., Evans, R., Pritzel, A., *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596,** 583–589. https://doi.org/10.1038/s41586-021-03819-2 (2021).
8. Baek, M., DiMaio, F., Anishchenko, I., *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373,** 871–876. https://doi.org/10.1126/science.abj8754 (2021).
9. Ahdritz, G., Bouatta, N., Floristean, C., *et al.* OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. http://dx.doi.org/10.1101/2022.11.20.517210 (2022).
10. Li, Z., Liu, X., Chen, W., *et al.* Uni-Fold: An Open-Source Platform for Developing Protein Folding Models beyond AlphaFold. http://dx.doi.org/10.1101/2022.08.04.502811 (2022).
11. Evans, R., O'Neill, M., Pritzel, A., *et al.* Protein complex prediction with AlphaFold-Multimer. https://doi.org/10.1101/2021.10.04.463034 (2021).
12. Zheng, W., Wuyun, Q., Li, Y., *et al.* Improving deep learning protein monomer and complex structure prediction using DeepMSA2 with huge metagenomics data. *Nature Methods* **21,** 279–289. ISSN: 1548-7105. http://dx.doi.org/10.1038/s41592-023-02130-4 (2024).
13. Krishna, R., Wang, J., Ahern, W., *et al.* Generalized biomolecular modeling and design with RoseTTAFold All-Atom. *Science.* ISSN: 1095-9203. http://dx.doi.org/10.1126/science.adl2528 (2024).
14. Qiao, Z., Nie, W., Vahdat, A., *et al.* State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nature Machine Intelligence* **6,** 195–208. ISSN: 2522-5839. http://dx.doi.org/10.1038/s42256-024-00792-z (2024).
15. Lin, Z., Akin, H., Rao, R., *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379,** 1123–1130. ISSN: 1095-9203. http://dx.doi.org/10.1126/science.ade2574 (2023).
16. Wu, R., Ding, F., Wang, R., *et al.* High-resolution de novo structure prediction from primary sequence. http://dx.doi.org/10.1101/2022.07.21.500999 (2022).

17. Fang, X., Wang, F., Liu, L., *et al.* A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. *Nature Machine Intelligence* **5,** 1087–1096. ISSN: 2522-5839. http://dx.doi.org/10.1038/s42256-023-00721-6 (2023).

18. Chowdhury, R., Bouatta, N., Biswas, S., *et al.* Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology* **40,** 1617–1623. ISSN: 1546-1696. http://dx.doi.org/10.1038/s41587-022-01432-w (2022).

19. Wang, W., Peng, Z. & Yang, J. Single-sequence protein structure prediction using supervised transformer protein language models. *Nature Computational Science* **2,** 804–814. ISSN: 2662-8457. http://dx.doi.org/10.1038/s43588-022-00373-3 (2022).

20. Wayment-Steele, H. K., Ojoawo, A., Otten, R., *et al.* Predicting multiple conformations via sequence clustering and AlphaFold2. *Nature* **625,** 832–839. ISSN: 1476-4687. http://dx.doi.org/10.1038/s41586-023-06832-9 (2023).

21. Chen, Y., Xu, Y., Liu, D., *et al.* SPIRED-Fitness: an end-to-end framework for the prediction of protein structure and fitness from single sequence. http://dx.doi.org/10.1101/2024.01.31.578102 (2024).

22. Hu, J., Wang, W. & Gong, H. Cerebra: a computationally efficient framework for accurate protein structure prediction. http://dx.doi.org/10.1101/2024.02.02.578551 (2024).

23. Mao, W., Ding, W., Xing, Y., *et al.* AmoebaContact and GDFold as a pipeline for rapid de novo protein structure prediction. *Nature Machine Intelligence* **2,** 25–33. https://doi.org/10.1038/s42256-019-0130-4 (2019).

24. Sillitoe, I., Bordin, N., Dawson, N., *et al.* CATH: increased structural coverage of functional space. *Nucleic Acids Research* **49,** D266–D273. ISSN: 1362-4962. http://dx.doi.org/10.1093/nar/gkaa1079 (2020).

25. Paszke, A., Gross, S., Chintala, S., *et al.* Automatic differentiation in PyTorch (2017).

26. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22,** 2577–2637. ISSN: 1097-0282. http://dx.doi.org/10.1002/bip.360221211 (1983).

27. Blondel, M., Teboul, O., Berthet, Q., *et al.* *Fast Differentiable Sorting and Ranking* in *INTERNATIONAL CONFERENCE ON MACHINE LEARNING, VOL 119* (eds Daume, H. & Singh, A.) **119.** International Conference on Machine Learning (ICML), ELECTR NETWORK, JUL 13-18, 2020 (2020).

28. Mariani, V., Biasini, M., Barbato, A., *et al.* lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* **29,** 2722–2728. https://doi.org/10.1093/bioinformatics/btt473 (2013).

29. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics* **57,** 702–710. https://doi.org/10.1002/prot.20264 (2004).

30. Spearman, C. The proof and measurement of association between two things. *International Journal of Epidemiology* **39,** 1137–1150. ISSN: 0300-5771. http://dx.doi.org/10.1093/ije/dyq191 (2010).

Supplementary Materials

GDFold2: a fast and parallelizable protein folding environment with freely defined objective functions

Tianyu Mi[1,2] and Haipeng Gong[1,2,*]

[1]MOE Key Laboratory of Bioinformatics, School of Life Sciences, Tsinghua University, Beijing, 100084, China

[2]Beijing Frontier Research Center for Biological Structure, Tsinghua University, Beijing, 100084, China

[*]To whom correspondence should be addressed: hgong@tsinghua.edu.cn

1 Supplementary Details for GDFold2

1.1 Statistical parameters used for the construction of statistical constraints

Table S1. Statistical values for scalars and vectors

Item	Value
l	1.33
d	3.80
θ_a	0.520
θ_b	0.449
$r_{\mathrm{C}_\alpha}, r_{\mathrm{C}}, r_{\mathrm{C}_\beta}, r_{CEN}$	1.70
r_{N}	1.55
r_{O}	1.52
r_{H}	1.20
$\mathbf{v}_{\mathrm{O}}$	[-0.6653, -1.0348, 0.0028]
$\mathbf{v}_{\mathrm{H}}$	[-0.4889, -0.8837, 0.0]

Table S2. Statistical internal coordinates of backbone atoms

Amino acid	C	N	C_β	*CEN*
GLY	[1.516, 0, 0]	[-0.574, 1.335, 0]	NA	NA
ALA	[1.524, 0, 0]	[-0.525, 1.361, 0]	[-0.533, -0.775, -1.196]	[-0.532, -0.775, -1.195]
CYS	[1.522, 0, 0]	[-0.521, 1.361, 0]	[-0.540, -0.570, -1.312]	[-1.268, -1.274, -1.474]
GLU	[1.524, 0, 0]	[-0.527, 1.360, 0]	[-0.525, -0.778, -1.208]	[-1.623, -1.894, -1.882]
ASP	[1.526, 0, 0]	[-0.524, 1.361, 0]	[-0.526, -0.779, -1.208]	[-1.258, -1.443, -1.455]
PHE	[1.523, 0, 0]	[-0.516, 1.363, 0]	[-0.528, -0.775, -1.208]	[-1.681, -1.818, -1.539]
ILE	[1.525, 0, 0]	[-0.492, 1.373, 0]	[-0.514, -0.795, -1.214]	[-1.331, -1.404, -1.601]
HIS	[1.522, 0, 0]	[-0.525, 1.360, 0]	[-0.523, -0.784, -1.208]	[-1.581, -1.688, -1.511]
LYS	[1.524, 0, 0]	[-0.524, 1.361, 0]	[-0.526, -0.778, -1.207]	[-1.849, -2.234, -1.984]
MET	[1.523, 0, 0]	[-0.523, 1.362, 0]	[-0.529, -0.771, -1.208]	[-1.620, -2.064, -1.667]
LEU	[1.523, 0, 0]	[-0.520, 1.363, 0]	[-0.535, -0.774, -1.211]	[-1.619, -1.857, -1.363]
ASN	[1.524, 0, 0]	[-0.535, 1.356, 0]	[-0.523, -0.800, -1.179]	[-1.318, -1.449, -1.392]
GLN	[1.524, 0, 0]	[-0.525, 1.361, 0]	[-0.517, -0.775, -1.215]	[-1.690, -1.947, -1.757]
PRO	[1.524, 0, 0]	[-0.568, 1.350, 0]	[-0.568, -0.637, -1.279]	[-1.296, 0.973, -1.528]
SER	[1.523, 0, 0]	[-0.530, 1.358, 0]	[-0.527, -0.765, -1.198]	[-0.698, -0.859, -1.706]
ARG	[1.524, 0, 0]	[-0.523, 1.362, 0]	[-0.558, -0.831, -1.146]	[-1.988, -2.416, -2.408]
THR	[1.524, 0, 0]	[-0.517, 1.363, 0]	[-0.506, -0.799, -1.215]	[-0.937, -1.026, -1.537]
TRP	[1.523, 0, 0]	[-0.522, 1.361, 0]	[-0.525, -0.774, -1.210]	[-1.500, -2.164, -1.648]
VAL	[1.524, 0, 0]	[-0.493, 1.373, 0]	[-0.513, -0.795, -1.215]	[-0.937, -1.304, -1.406]
TYR	[1.523, 0, 0]	[-0.519, 1.362, 0]	[-0.526, -0.776, -1.209]	[-1.757, -1.952, -1.625]

As described in **Methods**, atomic-level parameters summarized from high-resolution protein structures are used to construct statistical constraints. The parameters include peptide bond length (l), distance between adjacent residues (d), angles between adjacent covalent bonds within the peptide plane (θ_a and θ_b), van der Waals radii of backbone atoms ($r_{\mathrm{C}_\alpha}, r_{\mathrm{C}}, r_{\mathrm{N}}, r_{\mathrm{C}_\beta}, r_{CEN}, r_{\mathrm{O}}, r_{\mathrm{H}}$), as well as vectors of O and H atoms in the peptide plane coordinate system

($\mathbf{v}_{\mathrm{O}}$ and $\mathbf{v}_{\mathrm{H}}$), as listed in Table S1. The internal coordinates of backbone atoms of 20 common amino acids (Table S2) are calculated in the local coordinate system of each residue, taking C_{α} as origin ([0, 0, 0]).

1.2 GDFold2 is significantly more efficient in folding than pyRosetta

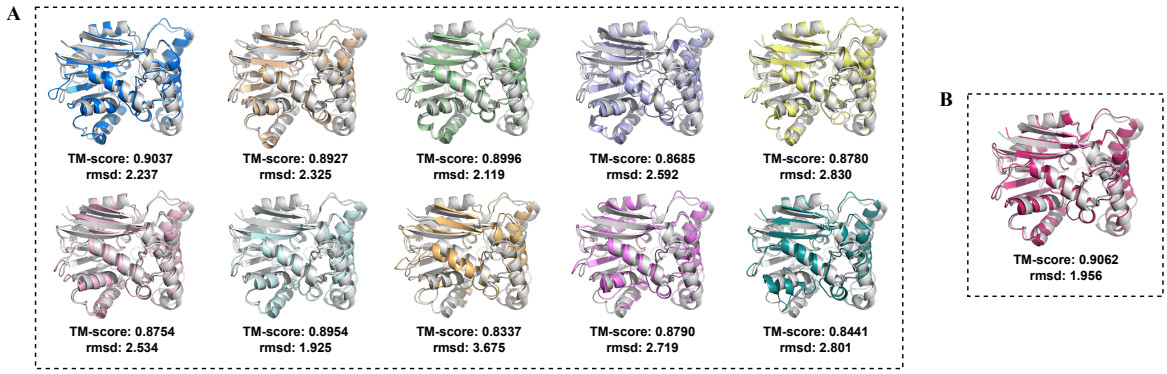
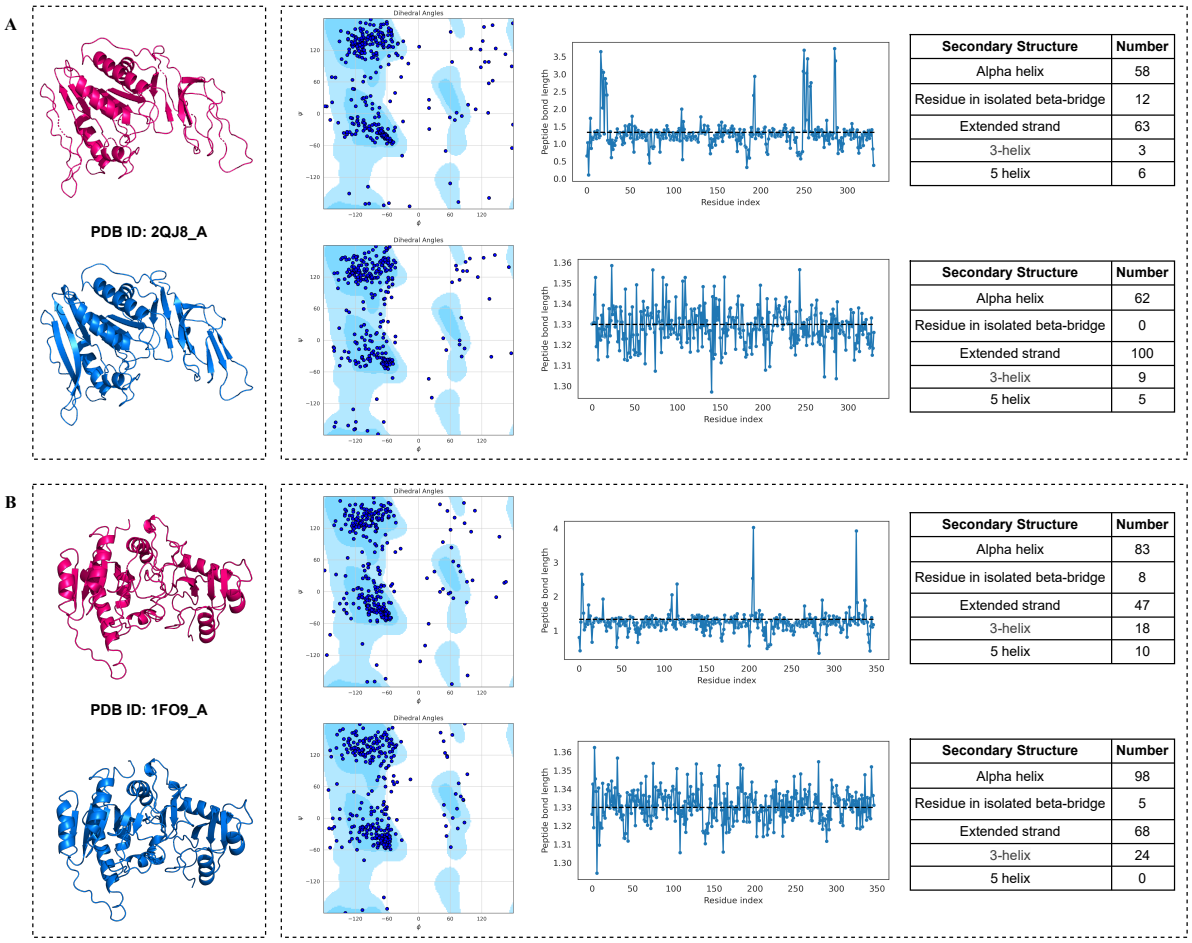


Figure S1. Comparison of the protein structures folded by GDFold2 and pyRosetta. A) Multiple prediction results output by GDFold2 through parallel folding (batch size = 10). **B)** Single prediction result output by pyRosetta. All of the generated structures are superimposed into the native structure (colored gray).

As shown in Figure S1A, multiple predicted protein structures are parallelly folded by GDFold2 using the RoseTTAFold predictions as constraints and are then relaxed following our modified pyRosetta-based FastRelax procedure. The total runtime to obtain all these prediction results is just 756.36 seconds. The single structure folded by pyRosetta with FastRela is generated by the same RoseTTAFold-predicted constraints (Figure S1B), which consumes 2614.89 seconds. The structural models generated by the two methods present the same level of quality. However, the folding efficiency of GDFold2 is much higher than that of the pyRosetta environment.

1.3 GDFold2 folding improves the local structural quality for Cerebra predictions



Secondary Structure	Number
Alpha helix	58
Residue in isolated beta-bridge	12
Extended strand	63
3-helix	3
5 helix	6

Secondary Structure	Number
Alpha helix	62
Residue in isolated beta-bridge	0
Extended strand	100
3-helix	9
5 helix	5

Secondary Structure	Number
Alpha helix	83
Residue in isolated beta-bridge	8
Extended strand	47
3-helix	18
5 helix	10

Secondary Structure	Number
Alpha helix	98
Residue in isolated beta-bridge	5
Extended strand	68
3-helix	24
5 helix	0

Figure S2. Comparison of structures generated from by GDFold2 and by direct construction based on Cerebra predictions. The comparison is shown for two cases: **A)** 2QJ8, chain A; **B)** 1FO9, chain A. In both cases, the structures directly output from Cerebra (red) and their properties are shown in the upper row, whereas the GDFold2-folded structures (blue) and their properties are shown in the lower row.

As shown in Figure S2, the incorrectly predicted local geometric information may lead to the appearance of local chain breakage at the peptide bonds (unrealistic distance between the adjacent C and N atoms), unreasonable distribution of main-chain torsion angles (violation of the Ramachandran plot) and damage on secondary structures in the raw Cerebra predictions. The protein structures generated by GDFold2 using Cerebra-predicted information and statistical constraints exhibit refinement in the local structural properties including the Ramachandran plot of main-chain torsion angles, the distribution of peptide-bond lengths as well as the number of residues in secondary structure contents (especially for categories of isolated beta-bridge and extended strand) based on DSSP calculation.

2 Supplementary Details for the QA Model

2.1 Case study for the decoy structures folded by GDFold2

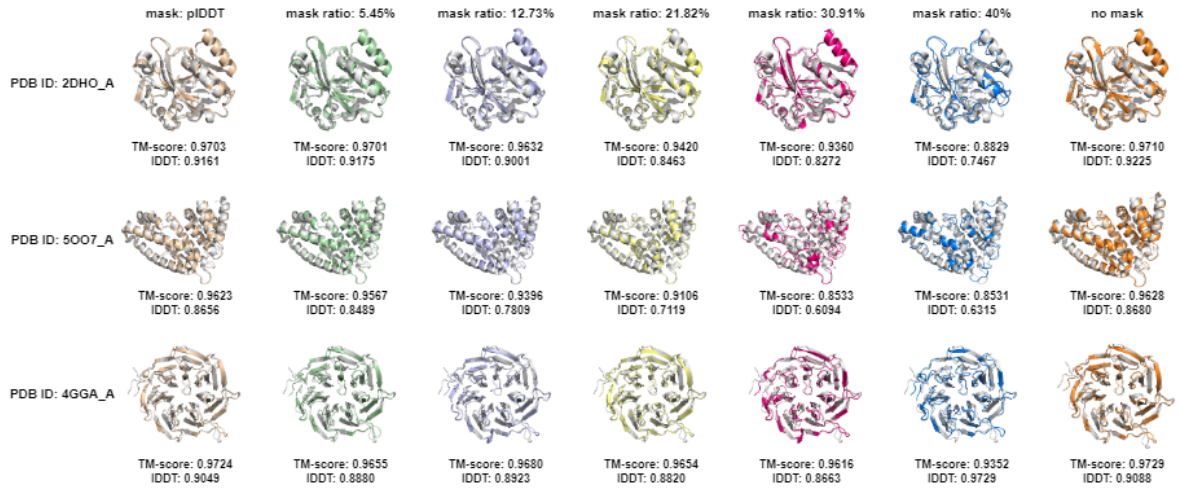


Figure S3. Decoy structures generated using geometric information predicted by SPIRED

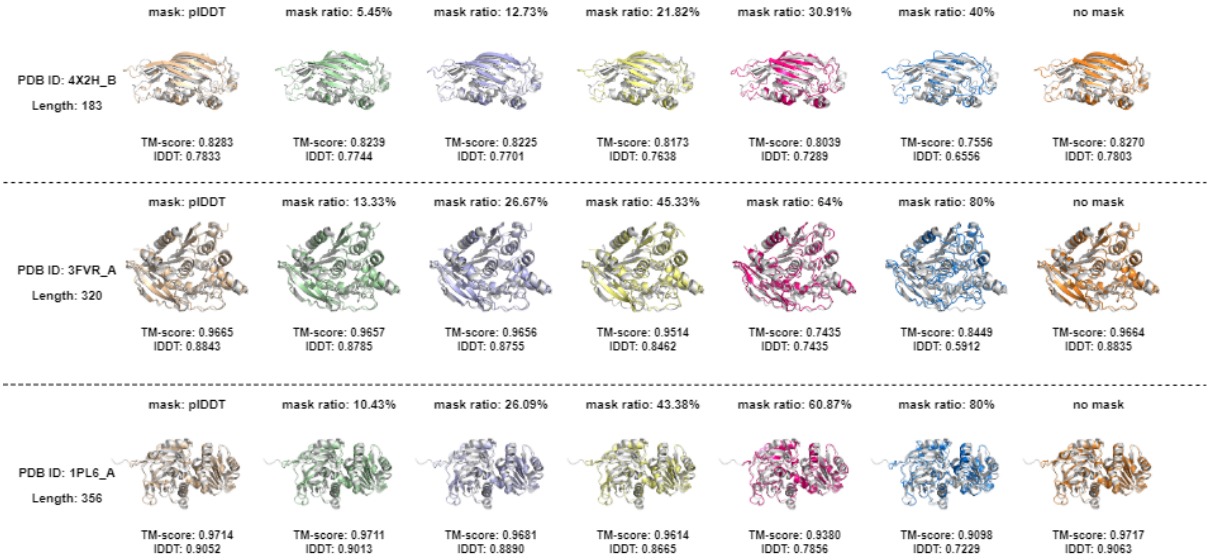


Figure S4. Decoy structures generated using geometric information predicted by Cerebra

As described in **Methods**, the decoy structures are obtained by multiplying various mask weights to predicted geometric information and converting the masked 2D information into 3D structures through GDFold2 in parallel. The quality of these structures decreases with the increase of the mask ratio. As shown in Figures S3 and S4, geometric information predicted from SPIRED is more sensitive to masking than that from Cerebra. Therefore, the upper limit of mask ratio is set as 0.4 for SPIRED and 0.8 for Cerebra.

2.2 QA model performs well on selecting the best prediction result

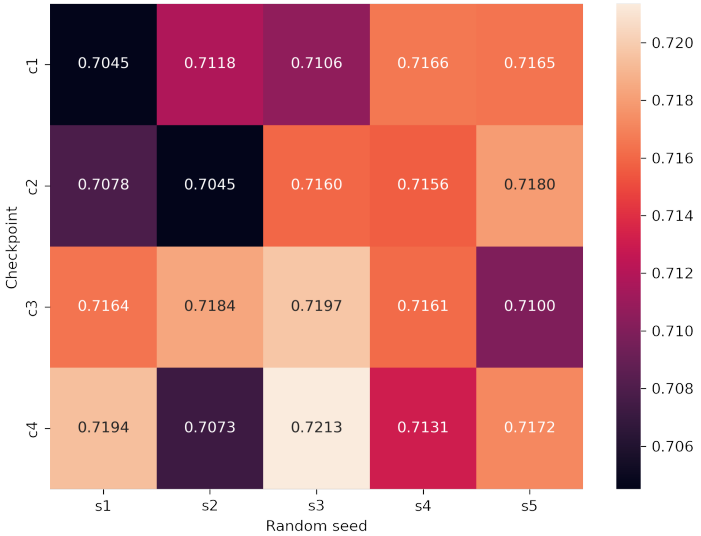


Figure S5. True evaluation score distribution of the test dataset

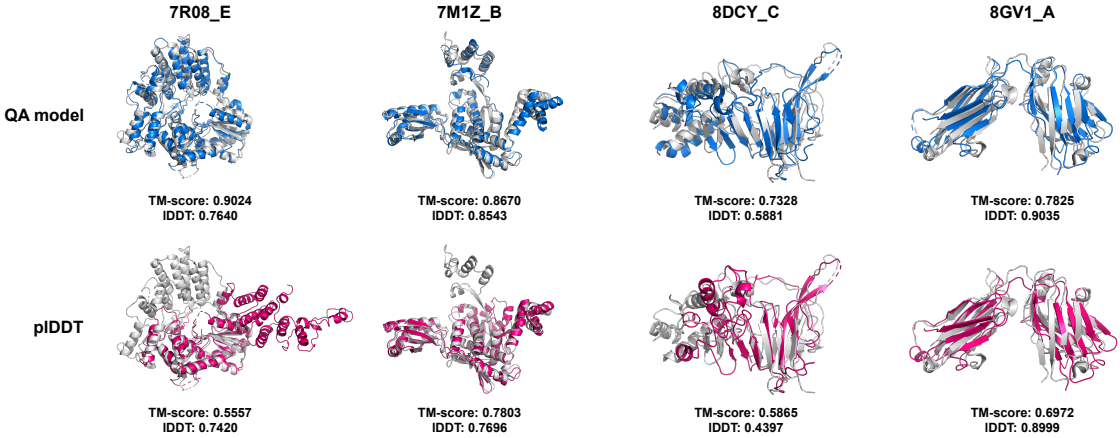


Figure S6. Comparison of the selected best structures

Figure S5 summarizes the average true evaluation scores of the 76 targets in CAMEO testing dataset folded by GDFold2 using geometric information predicted by 20 Cerebra models of different random seeds and checkpoints. We utilized the QA model to select the best one among the 20 structural models generated for each target protein and compared the entire results with those selected by plDDT. Based on **Results** and showcases in Figure S6, the structures chosen by the QA model have higher quality than those selected by plDDT in terms of TM-score, lDDT and the rationality upon visual inspection.

3 Algorithm and Pseudocode for GDFold2 and the QA model

Algorithm 1 Predicted vector constraint (Cerebra)

def **vector**($R \in \mathbb{R}^{b\times L\times 3\times 3}, I \in \mathbb{R}^{b\times 1\times L\times 3\times 4}, C_\alpha, C, N, C_\beta, Pred$): $C_\alpha, C, N, C_\beta \in \mathbb{R}^{b\times L\times 3},\ L = \text{Length}$

$i, r, t = \text{reshape}(Pred)$ $i = [a_1, a_2, ..., a_N]\,,\ r \in \mathbb{R}^{b\times N\times L\times 3\times 3},\ t \in \mathbb{R}^{b\times N\times L\times 3}$

$B = \text{stack}([C_\alpha, C, N, C_\beta]\,, 3) \quad \in \mathbb{R}^{b\times L\times 3\times 4}$

$l = r \otimes I + t$

$v = R_i^{-1} \otimes (B - C_{\alpha,i})$

$\mathcal{L}_{vec} = \frac{1}{b\times N\times L}\sum(\|v - l\|_2)$

return $\mathcal{L}_{vec}$

Algorithm 2 Predicted vector constraint (SPIRED)

def **vector**($C_\alpha, Pred$): $C_\alpha \in \mathbb{R}^{b\times L\times 3},\ L = \text{Length}$

$i, t = \text{reshape}(Pred)$ $i = [a_1, a_2, ..., a_N]\,,\ t \in \mathbb{R}^{b\times L\times 3}$

$l = \text{concat}(t_i)$

$C_{\alpha,ref} = \text{concat}(C_{\alpha,i})$

$v = C_\alpha - C_{\alpha,ref}$

$\mathcal{L}_{vec} = \frac{1}{b\times L}\sum(\|v - l\|_2)$

return $\mathcal{L}_{vec}$

Algorithm 3 Infer the internal coordinate of the previous C atom in each residual coordinate system

def **prev**($\phi \in \mathbb{R}^{L-2}, I_{C_\alpha}, I_C, I_N, l = 1.33, \theta_{right} = 1.5708, \theta_a = 0.520, \epsilon = 1e^{-6}$): $I_{C_\alpha}, I_C, I_N \in \mathbb{R}^{L\times 3}$

$b = \|I_N - I_{C_\alpha}\|_2$ (row-wise) $\quad \in \mathbb{R}^L$

$\theta' = \sum((N - C_\alpha) \times (C - C_\alpha)/(\|N - C_\alpha\|_2 \times \|C - C_\alpha\|_2))$ (row-wise) $\quad \in \mathbb{R}^L$

$\theta'' = \arccos(\theta') - \theta_{right}$

$v_1 = \theta_a \times l$

$v_2 = \sqrt{l^2 - v_1^2 + \epsilon}$

$x_i = v_2 \times \cos(\phi_i) \times \cos(\theta_i'') + \theta_i' \times (b_i + v_1)$ $i \in [2, ..., L-1]$

$y_i = v_2 \times \cos(\phi_i) \times \sin(\theta_i'') + \cos(\theta_i'') \times (b_i + v_1)$ $i \in [2, ..., L-1]$

$z_i = -\text{sign}(\phi_i) \times \sqrt{v_2^2 - (v_2 \times \cos(\phi_i))^2 + \epsilon}$

$C_p = \text{stack}([x, y, z], 2) \quad \in \mathbb{R}^{(L-2)\times 3}$

return C_p

Algorithm 4 Infer the internal coordinate of the next N atom in each residual coordinate system

def **next**($\psi \in \mathbb{R}^{L-2}, \theta_b = 0.449, l = 1.33, I_C^x \in \mathbb{R}^{L-2}, \epsilon = 1e^{-6}$):

1: $x_i = \theta_b \times l + I_C^x$
2: $y_i = \cos(\psi_i) \times \sqrt{l^2 - (\theta_b \times l)^2 + \epsilon}$ $\quad i \in [2, ..., L-1]$
3: $z_i = \text{sign}(\psi) \times \sqrt{l^2 - (\theta_b \times l)^2 - y_i^2 + \epsilon}$ $\quad i \in [2, ..., L-1]$
4: $N_p = \text{stack}([x, y, z], 2) \quad \in \mathbb{R}^{(L-2)\times 3}$
5: return N_p

Algorithm 5 Predicted dihedral angles constraint

def **dihedral**($R \in \mathbb{R}^{b\times(L-2)\times 3\times 3}, C_\alpha, C, N, \phi, \psi$): $\quad C_\alpha, C, N \in \mathbb{R}^{b\times(L-2)\times 3} \quad \phi, \psi \in \mathbb{R}^{L-2}$

1: $C_p = \text{reshape}\,(\text{prev}(\phi)) \quad \in \mathbb{R}^{b\times(L-2)\times 3\times 1}$
2: $N_p = \text{reshape}\,(\text{next}(\psi)) \quad \in \mathbb{R}^{b\times(L-2)\times 3\times 1}$
3: $C_l = \text{reshape}\,(R \odot C_p + C_\alpha) \quad \in \mathbb{R}^{b\times(L-2)\times 3}$
4: $N_l = \text{reshape}\,(R \odot N_p + C_\alpha) \quad \in \mathbb{R}^{b\times(L-2)\times 3}$
5: $\mathcal{L}_\phi = \|C - C_l\|_2 \quad \in \mathbb{R}^{b\times(L-2)}$
6: $\mathcal{L}_\psi = \|N - N_l\|_2 \quad \in \mathbb{R}^{b\times(L-2)}$
7: $\mathcal{L}_{dihe} = \frac{\mathcal{L}_\phi + \mathcal{L}_\psi}{b\times(L-2)}$
8: return $\mathcal{L}_{dihe}$

Algorithm 6 Peptide plane constraints

def **peptide**($C_\alpha^l, C_\alpha^r, C, N, l = 1.33, \theta_1 = 2.0281, \theta_2 = 2.1241, \omega = -1.0, m$): $C_\alpha^l, C_\alpha^r, C, N \in \mathbb{R}^{b\times(L-1)\times 3}$

peptide bond length constraint

$b_p = \|C - N\|_2$

$\mathcal{L}_f = \text{concat}(0, |b_p - l|)$

$\mathcal{L}_b = \text{concat}(|b_p - l|, 0)$

$\mathcal{L}_{bond} = \frac{\mathcal{L}_f + \mathcal{L}_b}{b\times L}$

peptide angles constraints

$v_1 = (C_\alpha^l - C)/\|C_\alpha^l - C\|_2$

$v_2 = (N - C)/\|N - C\|_2$

$\theta_{p1} = \arccos(v_1 \cdot v_2)$

$\mathcal{L}_{a1} = \frac{\sum|\theta_{p1} - \theta_1|}{b\times(L-1)}$

$v_3 = (C - N)/\|C - N\|_2$

$v_4 = (C_\alpha^r - C)/\|C_\alpha^r - C\|_2$

$\theta_{p2} = \arccos(v_3 \cdot v_4)$

$\mathcal{L}_{a2} = \frac{\sum|\theta_{p2} - \theta_2|}{b\times(L-1)}$

peptide plane constraint

$v_1 = (C_\alpha^l - C) \times (N - C)$

$v_2 = (C - N) \times C_\alpha^r - C)$

$\omega_p = \frac{v_1}{\|v_1\|_2} \cdot \frac{v_2}{\|v_2\|_2}$

$\mathcal{L}_\omega = m \times |\omega_p - \omega|$ $m \in \mathbb{R}^{b\times L}$

$\mathcal{L}_\omega += (1 - m) \times ||p_\omega| - |\omega||)$

return $\mathcal{L}_{bond} + \mathcal{L}_{a1} + \mathcal{L}_{a2} + \mathcal{L}_\omega$

Algorithm 7 C_α-C_α distance constraint

def **global**($C_\alpha, d = 3.80$):

$d_f = |C_{\alpha,i} - C_{\alpha,i+1}|$ $i \in [1, ..., L-1]$

$d_b = |C_{\alpha,i} - C_{\alpha,i-1}|$ $i \in [2, ..., L]$

$\mathcal{L}_f = \text{concat}(|d_f - d|, 0)$

$\mathcal{L}_b = \text{concat}(0, |d_b - d|)$

$\mathcal{L}_{dist} = \frac{\mathcal{L}_f + \mathcal{L}_b}{b\times L}$

return $\mathcal{L}_{dist}$

Algorithm 8 Van der Waals repulsive force constraint

def **vdw**($C_\alpha, C, N, C_\beta, O, H, CEN \in \mathbb{R}^{b\times L\times 3}, m, r, t = 1.20$): m is mask for bonded atom pairs

$A = \text{reshape}(\text{stack}([C_\alpha, C, N, C_\beta, O, H, CEN], 2)) \quad \in \mathbb{R}^{b\times(7\times L)\times 3}$

$A_1 = \text{reshape}(A) \quad \in \mathbb{R}^{b\times(7\times L)\times 1\times 3}$

$A_2 = \text{reshape}(A) \quad \in \mathbb{R}^{b\times 1\times(7\times L)\times 3}$

$d = \|A_1 - A_2\|_2$

$\mathcal{L} = \sum \min(m \times (d - r - t), 0)$ r is Van der Waals radius for all atom pairs

$\mathcal{L}_{vdw} = \frac{\mathcal{L}}{b}$

return $\mathcal{L}_{vdw}$

Algorithm 9 Convert Euler Angle into rotation matrix

def **theta2mat**($\theta_x, \theta_y, \theta_z$): $\theta_x, \theta_y, \theta_z \in \mathbb{R}^{b\times L\times 3}$

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix}$$

$$R_y = \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix}$$

$$R_z = \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$R = R_x \otimes R_y \otimes R_z$

return R $R \in \mathbb{R}^{b\times L\times 3\times 3}$

Algorithm 10 Calculate the rotation matrix of residual coordinate system

def **coords2rot**(a, b, c): $a, b, c \in \mathbb{R}^{b\times(L-1)\times 3}$

$x = (a - b)/\|a - b\|_2$

$z = x \times (c - b)/\|x \times (c - b)\|_2$

$y = x \times y$

$A = \text{stack}([x, y, z], 2) \quad \in \mathbb{R}^{b\times(L-1)\times 3\times 3}$

$U, S, V = \text{SVD}(A)$

$r = V^{-1} \otimes U^{-1}$

return r $r \in \mathbb{R}^{b\times(L-1)\times 3\times 3}$

Algorithm 11 Infer the global coordinates of oxygen atoms

def **oxygen**$(R_L, I_C, C_\alpha, C, N, v_o, v_l)$: $R_L \in \mathbb{R}^{b\times3\times3},\ I_C \in \mathbb{R}^{b\times3},\ C_\alpha, C, N \in \mathbb{R}^{b\times(L-1)\times3}$

p = coords2rot(N, C, C_α)
$o = p \otimes v_o + C$
$o_l = R_L \otimes (I_C + v_l) + C_{\alpha,L}$
$o \leftarrow$ concat(o, o_l)
return o $\quad o \in \mathbb{R}^{b\times L\times3}$

Algorithm 12 Infer the global coordinates of hydrogen atoms

def **hydrogen**$(R_1, I_N, C_\alpha, C, N, v_h, v_f)$: $R_1 \in \mathbb{R}^{b\times3\times3},\ I_N \in \mathbb{R}^{b\times3},\ C_\alpha, C, N \in \mathbb{R}^{b\times(L-1)\times3}$

p = coords2rot(C, N, C_α)
$h = p \otimes v_h + N$
$h_f = R_1 \otimes (I_N + v_f) + C_{\alpha,1}$
$h \leftarrow$ concat(h_f, h)
return h $\quad h \in \mathbb{R}^{b\times L\times3}$

Algorithm 13 Calculate the coordinates of backbone atoms and the rotations of residual coordinate systems

def **info**$(C_\alpha, \theta, I_{C_\alpha}, I_C, I_N, I_{C_\beta}, I_{CEN}, I_O, I_H, oh)$: $C_\alpha, \theta \in \mathbb{R}^{b\times L\times3}$

R = theta2mat (θ)
$C_\alpha, C, N, C_\beta, CEN = R \otimes (I_{C_\alpha}, I_C, I_N, I_{C_\beta}, I_{CEN})$
if $oh = True$ **then**
    O = oxygen$(R_L, I_C, C_\alpha, C, N)$
    H = hydrogen$(R_1, I_N, C_\alpha, C, N)$
    return $R, [C_\alpha, C, N, C_\beta, CEN, O, H]$
else
    return $R, [C_\alpha, C, N, C_\beta, CEN]$
end if

Algorithm 14 Protein folding process

```
def fold(pred, params):
I_{C_α}, I_C, I_N, I_{C_β}, I_{CEN} = params
for e = 0 → S + 800 do
    C_α, θ = Model(e)
    if e < S then
        R, [C_α, C, N, C_β, CEN] = info(C_α, θ, oh = False)
    else
        R, [C_α, C, N, C_β, CEN, O, H] = info(C_α, θ, oh = True)
    end if
    L_vec = vector(R, C_α, C, N, C_β, pred, params)
    L_dihe = dihedral(R, C_α, C, N, pred, params)
    if e < S then
        L = L_vec + L_dihe
    else
        L_extra = peptide(C_α, C, N)+ global(C_α)
        if e < S + 400 then
            L = L_vec + L_dihe + L_extra
        else
            L_vdw = vdw(C_α, C, N, C_β, CEN)
            if e < S + 600 then
                L = L_vec + L_dihe + L_extra + 10 × L_vdw
            else
                L = L_vec + L_dihe + L_extra + 5 × L_vdw
            end if
        end if
    end if
    Model ← Backward(L)
end for
return b_i
```

Algorithm 15 Graph Vector Perceptron

def **GVP**($\mathbf{s}, \mathbf{V}$): $(\mathbf{s}, \mathbf{V}) \in \mathbb{R}^n \times \mathbb{R}^{\nu \times 3}$

$h \leftarrow max(\nu, \mu)$

$\mathbf{V}_h \leftarrow \mathbf{W}_h \mathbf{V} \quad \in \mathbb{R}^{h \times 3}$

$\mathbf{V}_\mu \leftarrow \mathbf{W}_\mu \mathbf{V}_h \quad \in \mathbb{R}^{\mu \times 3}$

$\mathrm{s}_h \leftarrow \|\mathbf{V}_h\|_2$ (row-wise) $\quad \in \mathbb{R}^h$

$\mathrm{v}_\mu \leftarrow \|\mathbf{V}_\mu\|_2$ (row-wise) $\quad \in \mathbb{R}^\mu$

$\mathrm{s}_{h+n} \leftarrow \mathrm{concat}(\mathrm{s}_h, \mathrm{s}) \quad \in \mathbb{R}^{h+n}$

$\mathrm{s}_m \leftarrow \mathbf{W}_m s_{h+n} + \mathbf{b} \quad \in \mathbb{R}^m$

$\mathbf{s}' \leftarrow \sigma(s_m) \quad \in \mathbb{R}^m$

$\mathbf{V}' \leftarrow \sigma^+(\mathrm{v}_\mu) \odot \mathbf{V}_\mu$ (row-wise multiplication) $\quad \in \mathbb{R}^{\mu \times 3}$

return $(\mathbf{s}', \mathbf{V}')$ $(\mathbf{s}', \mathbf{V}') \in \mathbb{R}^m \times \mathbb{R}^{\mu \times 3}$