DeepPSC (protein structure camera): computer vision-based protein backbone structure reconstruction from alpha carbon trace as a case study

Xing Zhang[1], Junwen Luo[1], Yi Cai[1], Wei Zhu[1], Xiaofeng Yang[1]*, Hongmin Cai[2]*, and Zhanglin Lin[1]*

[1]School of Biology and Biological Engineering, [2]School of Computer Science and Engineering, South China University of Technology, 382 East Outer Loop Road, University Park, Guangzhou 510006, China;

* To whom correspondence should be addressed:

[1]School of Biology and Biological Engineering, South China University of Technology, 382 East Outer Loop Road, University Park, Guangzhou 510006, China; Tel: +86 (20) 3938-0680; Fax: +86 (20) 3938-0601; Email: zhanglinlin@scut.edu.cn (Z.L.); biyangxf@scut.edu.cn (X.Y.).

[2]School of Computer Science and Engineering, South China University of Technology, 382 East Outer Loop Road, University Park, Guangzhou 510006, China; Tel: +86 (20) 3938-2850; Fax: +86 (20) 3938-2850; Email: hmcai@scut.edu.cn (H.C.).

Abstract

Deep learning has been increasingly used in protein tertiary structure prediction, a major goal in life science. However, all the algorithms developed so far mostly use protein sequences as input, whereas the vast amount of protein tertiary structure information available in the Protein Data Bank (PDB) database remains largely unused, because of the inherent complexity of 3D data computation. In this study, we propose Protein Structure Camera (PSC) as an approach to convert protein structures into images. As a case study, we developed a deep learning method incorporating PSC (DeepPSC) to reconstruct protein backbone structures from alpha carbon traces. DeepPSC outperformed all the methods currently available for this task. This PSC approach provides a useful tool for protein structure representation, and for the application of deep learning in protein structure prediction and protein engineering.

Introduction

Protein structure determination is an ongoing issue and a major goal in life science that has captivated the attention of scientists for decades. Experimentally, protein structures have been mostly determined by X-ray diffraction crystallography[1], and to a less extent by nuclear magnetic resonance spectroscopy[2]. In recent years, cryo-electron microscopy (EM) has also been increasingly used for structure determination[3]. As an alternative to experimental methods, computational methods have also been developed for predicting protein structures from protein sequences, and deep learning has recently been applied to this prediction problem[4-6]. In particular, DeepMind proposed a method called AlphaFold[4], which significantly outperformed all previous prediction methods. A number of algorithms that extract features from protein primary sequences for the purpose of protein function prediction and protein engineering, *e.g.*, UniRep[7] and TAPE[8], represent a further advancement in the field. Other applications of deep learning include protein fold recognition[9], and the predictions of protein secondary structures[10], protein functions[11], and drug protein interactions[12].

However, all the deep learning methods developed so far utilize only protein sequences as input, whereas the vast amount of protein tertiary structure information available in the rapidly expanding PDB database has not been sufficiently exploited in the calculations, due to its complexity. There are presently three common coarse approximation approaches for protein structure representation, namely, k nearest residues[13], distance or contact maps[14], and 3D grids[15], but all with limited utility. Thus, we are interested in the following question: how to utilize protein structure information

in deep learning?

It is well known that images can be efficiently processed by deep learning, and particularly in recent years, convolutional neural networks (CNN) have been successfully used in an array of computer vision tasks such as image classification[16], object detection[17], and face recognition[18]. CNN can understand an object in the Euclidean space by extracting visual features from the corresponding image[19,20]. Fig. 1a shows a typical workflow of computer vision-based image classification. Here we

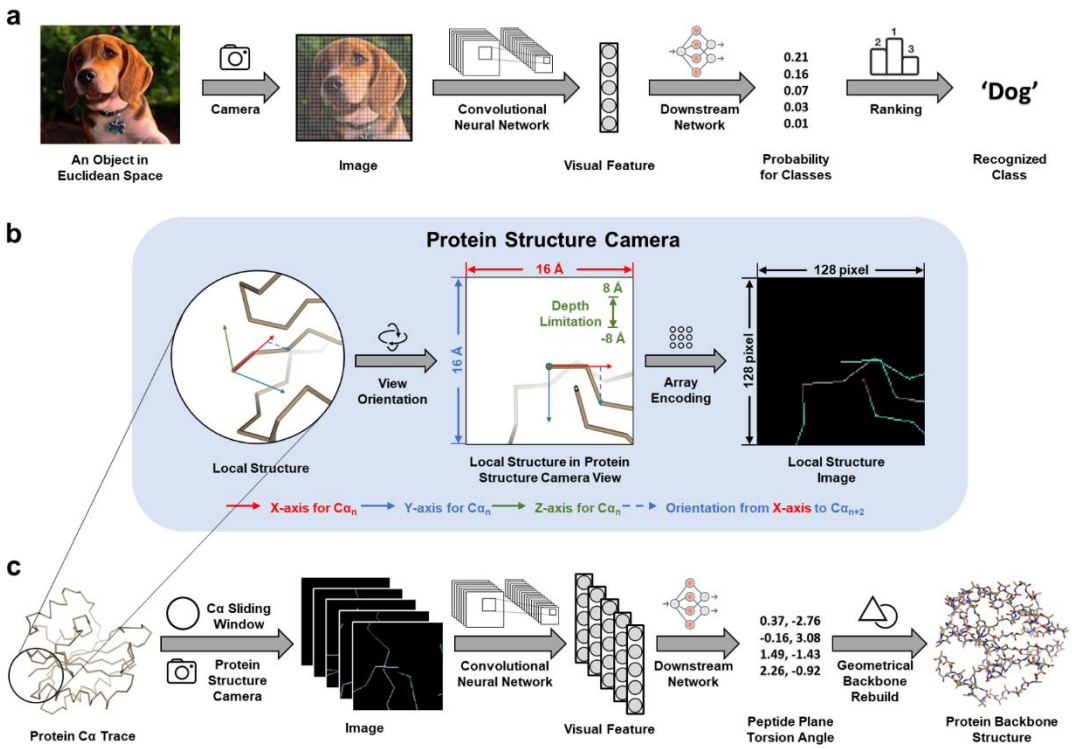


Figure 1. Schematic for the deep learning algorithm used in this work, or DeepPSC. a) The workflow of a typical computer vision task. b) Visualization of Protein Structure Camera (PSC) workflow. This figure is generated with the Chimera software [50]. c) The workflow of the deep learning-based algorithm DeepPSC used in this work for reconstructing backbone structures from alpha-carbon traces.

propose a "Protein Structure Camera" (PSC) approach for converting protein tertiary structures into images for computer vision processing. In PSC, we used a 16 Å × 16 Å × 16 Å sliding cubic window centered on the alpha carbons of the amino acid residues (Cα) to dissect a protein structure (Fig. 1b). This was then turned into a group of compressed two dimensional 16 Å× 16 Å images with a -8 Å to 8 Å depth range, which were then fed into a CNN and implemented into a deep learning-based network architecture, or DeepPSC (Fig. 1c).

As a case study, we applied this DeepPSC for reconstructing protein backbone structures (containing atoms C, N, O, Cβ in addition to Cα) based on Cα traces, which is an important task for protein structure determination by experimental means and for protein structure prediction by computational approaches. Several protein structure refinement methods have been developed for the analysis of EM images to generate high-quality structure models, such as PHENIX[21] and Coot[22]. Within these algorithms, the positions of the Cα, which are the atoms that can be located with the highest accuracy, are determined first. Subsequently the backbone structure and then the full atom model are generated. Similarly, many computational algorithms predict the Cα trace as a preliminary reduced model. PHENIX ensembles PULCHRA[23] for backbone reconstruction, which uses a simple force field and steepest descent minimization. Coot ensembles CALPHA[24,25], which is based on a library of backbone fragments compiled from experimentally determined structures. The widely used computational protein structure prediction platform I-TASSER[26] ensembles REMO[27], which directly reconstructs full-atom models (including the backbones) from a backbone isomer

library. Similar library-based methods include BBQ[28], SABBAC[29], and PD2[30], which often achieve better performance than PULCHRA or REMO. These three backbone structure reconstruction methods have also been applied for experimental structure determination[31], although they have not been incorporated in PHENIX or Coot. A significant limitation of the library-based methods, however, is that the wide range of conformations of protein backbones cannot be sufficiently represented by the limited number of fragments in the libraries.

In this work, we found that our DeepPSC approach outperformed all the previously reported methods for backbone reconstruction, including the benchmark PD2, and the ablation tests showed that the visual feature extracted from the protein structure images provided the main contribution for the improved performance.

Results

Represent Cα trace as images by protein structure camera

The PSC concept is shown in Fig. 1b. Given an Cα-trace $\{C\alpha_1, C\alpha_2, \dots, C\alpha_L\}$, where $C\alpha_n \in \mathbb{R}^3$ is the coordinate of the n^{th} Cα atom and L is the number of residues, PSC represents it as L images. Any given structural segment having $C\alpha_n$ as the center requires a preset orientation and scale. We defined the orientation from $C\alpha_n$ to $C\alpha_{n+1}$ as the X-axis. The Y-axis was then determined by the orientation from the X-axis to $C\alpha_{n+2}$, and the Z-axis was defined such as to build a left-hand Cartesian coordinate on the given local structural segment. We set the orientation directed from the positive to

the negative regions of the Z-axis as the PSC view so that the orientations of different local structural segments could be normalized. For $C\alpha_{L-1}$, the position on the Y-axis was determined by the orientation from the X-axis to $C\alpha_{L-2}$. Similarly, for the last Cα, we defined the orientation from $C\alpha_L$ to $C\alpha_{L-1}$ as the X-axis and the orientation from the X-axis to $C\alpha_{L-2}$ as the Y-axis. An enlarged view of Fig. 1b is given as Supplementary Fig. 1.

Since 8 Å is generally regarded as the interaction distance cutoff between two residues[32], we used a sliding cubic window with a side length of 16 Å centered on the Cα, and a depth ranging from -8 Å to 8 Å was applied to the PSC view. Each PSC view was then encoded as an image with five channels, representing the Z-axis depth, the relative sequence position, and three key amino acid properties including hydrophobicity[33], bulkiness[34] and flexibility[35], respectively. The resolution of the image is 128 × 128 pixels. In the image, each Cα was first encoded as a pixel, and a straight line was used to connect adjacent Cα pixels. The values of the properties along the straight line were interpolated from the two Cα pixel values. A given protein Cα trace was thus converted into a group of local structural images.

Present protein backbone structure as peptide plane torsion angles

In this study, we represented the structure of the protein backbone as peptide plane torsions, as reported in a previous study[36]. For convenience, we denoted the C atom and N atom in the n^{th} of all $L-1$ peptide planes by $C_n \in \mathbb{R}^3$ and $N_n \in \mathbb{R}^3$, respectively. Note that N_n, the N atom in the n^{th} peptide, is actually the $(n+1)^{th}$

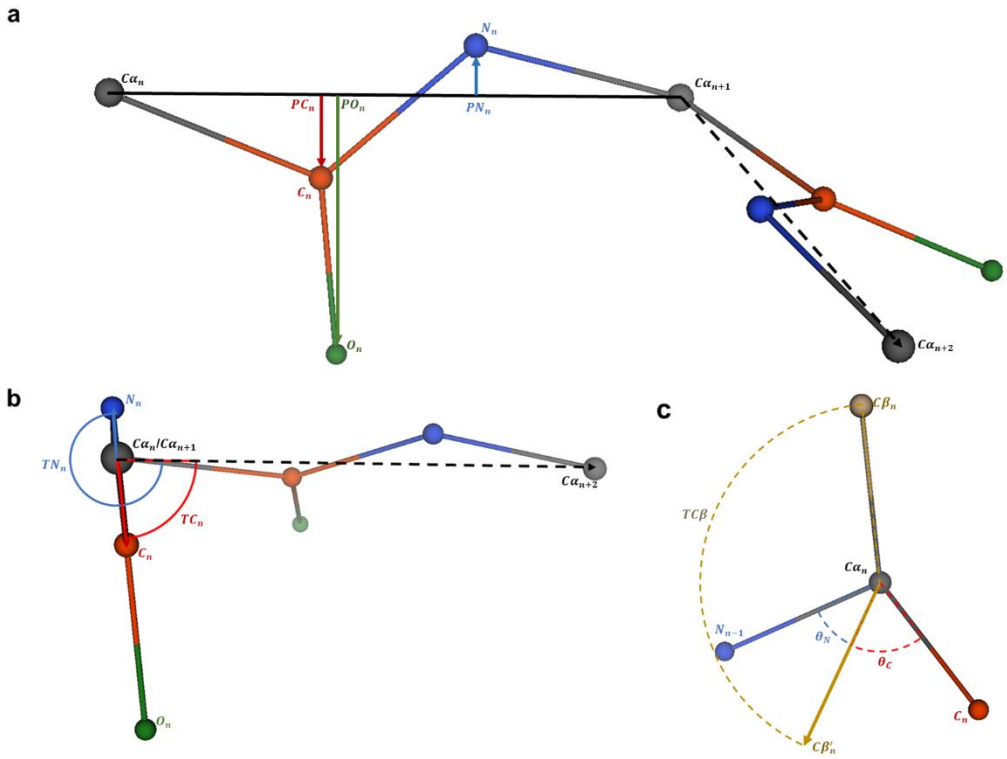


Figure 2. Diagrams of the backbone structure representation and rebuilding (only the case of a trans peptide plane is shown). a) Typical peptide plane conformation. $C\alpha_n$, C_n, O_n, N_n, $C\alpha_{n+1}$ located on the n^{th} peptide plane of the protein structure. PC_n, PN_n, and PO_n are the projections of C_n, N_n, and O_n on $\overrightarrow{C\alpha_n C\alpha_{n+1}}$, respectively. b) Side view from $C\alpha_n$ to C_{n+1}, in which TC_n and TN_n are the torsion angles from $C\alpha_{n+2}$ to C_n, and from $C\alpha_{n+2}$ to N_n, respectively, with $\overrightarrow{C\alpha_n C\alpha_{n+1}}$ as the axis. c) Rebuilding process for $C\beta_n$, using the constraint $\theta_N = \theta_C$, fixed bond length $\left|\overrightarrow{C\alpha_n C\beta_n}\right|$, and fixed torsion angle $TC\beta$.

N atom of the protein backbone. Besides, we defined the vectors from atom A to atom B as $\overrightarrow{AB} \equiv B - A$, with the corresponding unit vector being $\widehat{AB} \equiv \overrightarrow{AB} / \left|\overrightarrow{AB}\right|$. We applied a constraint that assumed $C\alpha_n$, C_n, O_n, N_n, $C\alpha_{n+1}$ forming a standard peptide plane (*trans* or *cis*). Within the given n^{th} *trans* peptide plane, as shown in

Fig.2a, the locations of C_n, O_n, N_n on the plane can be determined with a group of fixed lengths. For example, since $|C\alpha_n PC_n|$ and $|PC_n C_n|$ are fixed, we could locate C_n on the plane. Next, as shown in Fig. 2b, we used $\overrightarrow{C\alpha_{n+1}C\alpha_{n+2}}$ as a reference orientation to determine the torsion angles for each n^{th}peptide plane, *i.e.*, the torsion angle from $\overrightarrow{C\alpha_{n+1}C\alpha_{n+2}}$ to $\overrightarrow{C\alpha_n C_n}$ (TC_n), and that from $\overrightarrow{C\alpha_{n+1}C\alpha_{n+2}}$ to $\overrightarrow{C\alpha_n N_n}$ (TN_n), where TN_n is approximately 180 degrees larger than TC_n. The locations of $C\alpha_n$, C_n, O_n, N_n, $C\alpha_{n+1}$ were determined from the combination of all this information. In the case of *cis* peptides, the fixed lengths are different from those of *trans* peptides, and TN_n is close to TC_n. In all cases, TO_n is very close to TC_n, therefore TC_n was used as an approximation of TO_n in all calculations. TC_n and TN_n were encoded in the form of sine and cosine in the final representations.

Since the residues in proteins are L-amino acids, the coordinates of $C\beta_n$ can be determined when N_{n-1}, $C\alpha_n$, and C are known. Fig. 2c shows the rebuilding process used in this work. We first set the position of the projection of $C\beta_n(C\beta'_n)$ along the direction of the bisector of $\angle N_{n-1}C\alpha_n C_n$, with a fixed bond length $|C\alpha_n C\beta_n|$. By rotating $\overrightarrow{C\alpha_n C\beta'_n}$ with the fixed angle $TC\beta$, the location of $C\beta_n$ was determined. Without constraints from the peptide plane, the first N atom and the last C and O atoms of a protein backbone are usually highly flexible, therefore our method could not predict the positions of these atoms.

Specific geometrical calculations for all the above representations and rebuilding are provided in Methods.

Develop a deep neural network for PSC images

As shown in Fig. 3, the deep neural network implemented in DeepPSC takes local structure images as the input, and calculate the peptide plane torsions as the output. We first adopted ResNet50[37], the most used convolutional neural network for computer vision processing, to extract visual features from the images (Supplementary Fig. 2), which are labelled as "local structural features". Then, we used a bidirectional long short-term memory module (Bi-LSTM)[38,39], the most used recurrent neural network

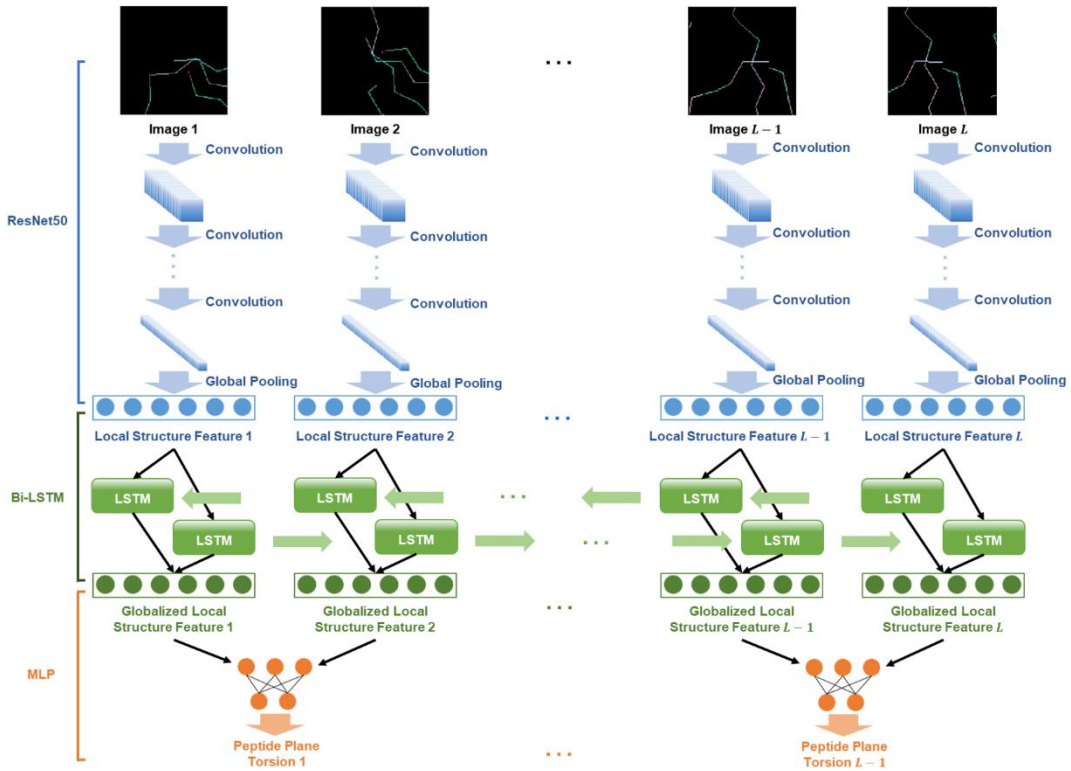


Figure 3. Network Architecture of DeepPSC. ResNet50 was used to extract visual features from images as local structure features. All local structure features were then fed into a Bi-LSTM module for information globalization among residues, yielding globalized local structure features. Finally, an MLP module was used to predict peptide plane torsions for pairs of the above adjacent globalized local structure features.

module for sequence modelling, to sequentially pass information between the extracted local structural features (Supplementary Fig. 3a). The outputs of this module were expected to mainly represent the local structures but they also contain sequential context information, and are labeled as "globalized local structural features". Afterward, considering that a single peptide plane is constructed by two adjacent residues, we paired every globalized local structural feature with the next one in the amino acid sequence as the "peptide plane feature". Finally, we used a multilayer perceptron (MLP)[40], a typical neural network module, to predict peptide plane torsions from the peptide plane features (Supplementary Fig. 3b).

To compare our DeepPSC method with previously reported protein structure representation methods, we additionally built two baseline methods. In the first baseline, we used the k nearest residues method[13] to represent the Cα trace, and to encode the network input. To maintain the input information as close to that of our method as possible, we enriched the representation by adding relative protein positions and residue properties. For this baseline, an MLP module (Supplementary Fig. 4) was used instead of ResNet50, to extract local structural features for the input format, since the latter cannot process this baseline input[13]. For the second baseline, protein structures were represented as distance maps and processed with a CNN (Supplementary Fig. 5), as previously reported, without any modifications[14].

DeepPSC outperforms other standard backbone reconstruction methods

We performed the 10-fold cross validation process on the three network architectures

208 **Table 1**. Overview of the results for various backbone reconstruction methods.

Methods	Models	Mean $RMSD_{100}$ (Å)	Mean GDT_P0.2 (%)	Mean RAMA outliers (%)
Rebuilt from PDB	——	0.040	95.76	0.22
DeepPSC	Ensemble model	**0.076**	**88.18**	**0.23**
	Single model	0.079±0.001	87.87±0.08	0.22±0.03
Baseline 1 (*k* nearest residues)	Ensemble model	0.101	83.44	0.50
	Single model	0.108±0.001	82.16±0.25	0.71±0.07
Baseline 2 (distance map)	Ensemble model	0.289	54.58	5.23
	Single model	0.289±0.001	54.42±0.48	5.30±0.17
PD2	——	0.149	73.26	0.90
BBQ	——	0.156	71.82	3.03
SABBAC*	——	0.201	58.30	1.94
PULCHRA	——	0.221	52.15	2.20

209 *SABBAC failed to process one of the 21 structures in the test set. The results shown

210 here were obtained with the other 20 test structures.

211

212 (DeepPSC, and the two baselines), and obtained 10 models for each architecture, for a

213 total of 30 models. Next, we applied each of these models to the test set and obtained

214 the predicted torsion angles as outputs. Subsequently, for each architecture, we took the

215 average of the outputs of the 10 cross validation models as the "ensemble model". Then
216 the outputs of the ten models and that of the ensemble model were used to rebuild the
217 backbone structures together with the corresponding Cα traces, and these rebuilt models
218 were evaluated with the three performance criteria, $RMSD_{100}$, GDT_P0.2, and RAMA
219 outliers (Table 1). The average performance of the ten models for each architecture was
220 calculated and shown as the "single model" performance, with the standard deviation
221 of the single model performance indicating the robustness of the network architecture.
222 Finally, we compared the performance of these architectures to that of PD2, BBQ,
223 SABBAC and PULCHRA (Table 1).

224 Generally, protein structures with resolution smaller than 2.0 Å are regarded as
225 high-quality structures[30]. According to the official statistics of PDB, up to July 27, 2020,
226 the median resolution of X-ray crystallography structures in the database is 2.03 Å. For
227 a typical 2.0 Å crystallographic model, the average error on atomic coordinates is lower
228 than 0.2 Å[41]. Therefore, we considered 0.2 Å as the benchmark in our performance
229 evaluation. Accordingly, we set the GDT cutoff at 0.2 Å to calculate the percentage of
230 atoms that can be regarded as acceptable in a high-quality structure.

231 Based on the mean $RMSD_{100}$, and the GDT_P0.2 and RAMA outliers percentages
232 as shown in Table 1, the backbone structures predicted by the ensemble model obtained
233 with DeepPSC clearly outperformed those predicted by the baseline methods as well as
234 the various traditional methods (PD2, BBQ, SABBAC and PULCHRA), in all three
235 criteria. In particular, the performance of baseline 1, which was devoid of the image
236 features of DeepPSC, suggested that the visual features extracted in DeepPSC were the

main factor for its improved performance. By comparing the results for the Rebuilt model (directly from the PDB) and the ensemble model of DeepPSC, it could be deduced that the deviations observed in DeepPSC consisted of two elements: (i) the first was represented by the deviations introduced during the rebuilding process per se, which were the deviations between the ideal peptide plane conformations and the experimentally determined peptide plane conformations; (ii) the second is represented by the deviations induced by the model fitting in DeepPSC. Therefore, future developments should focus on devising an alternative strategy in lieu of the peptide plane assumption.

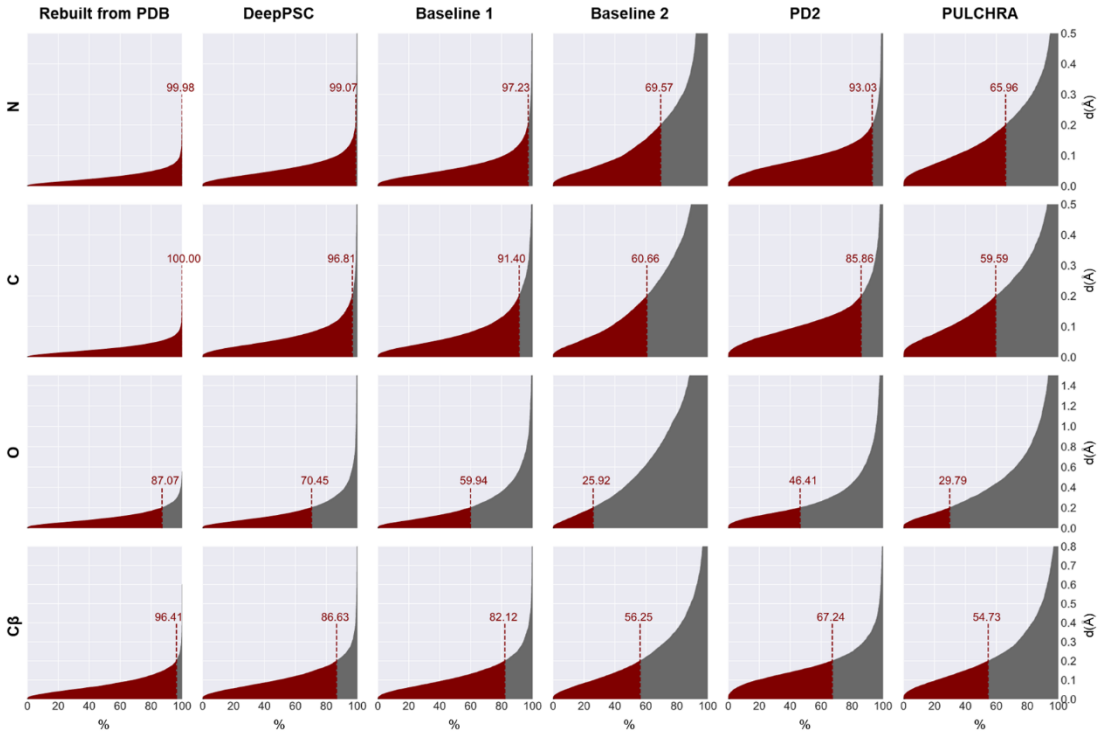


Figure 4. Distributions of the atomic coordinate deviations (rows) of the various reconstruction methods (columns). The GDT scores for 0.2 Å cutoff are indicated in the plots.

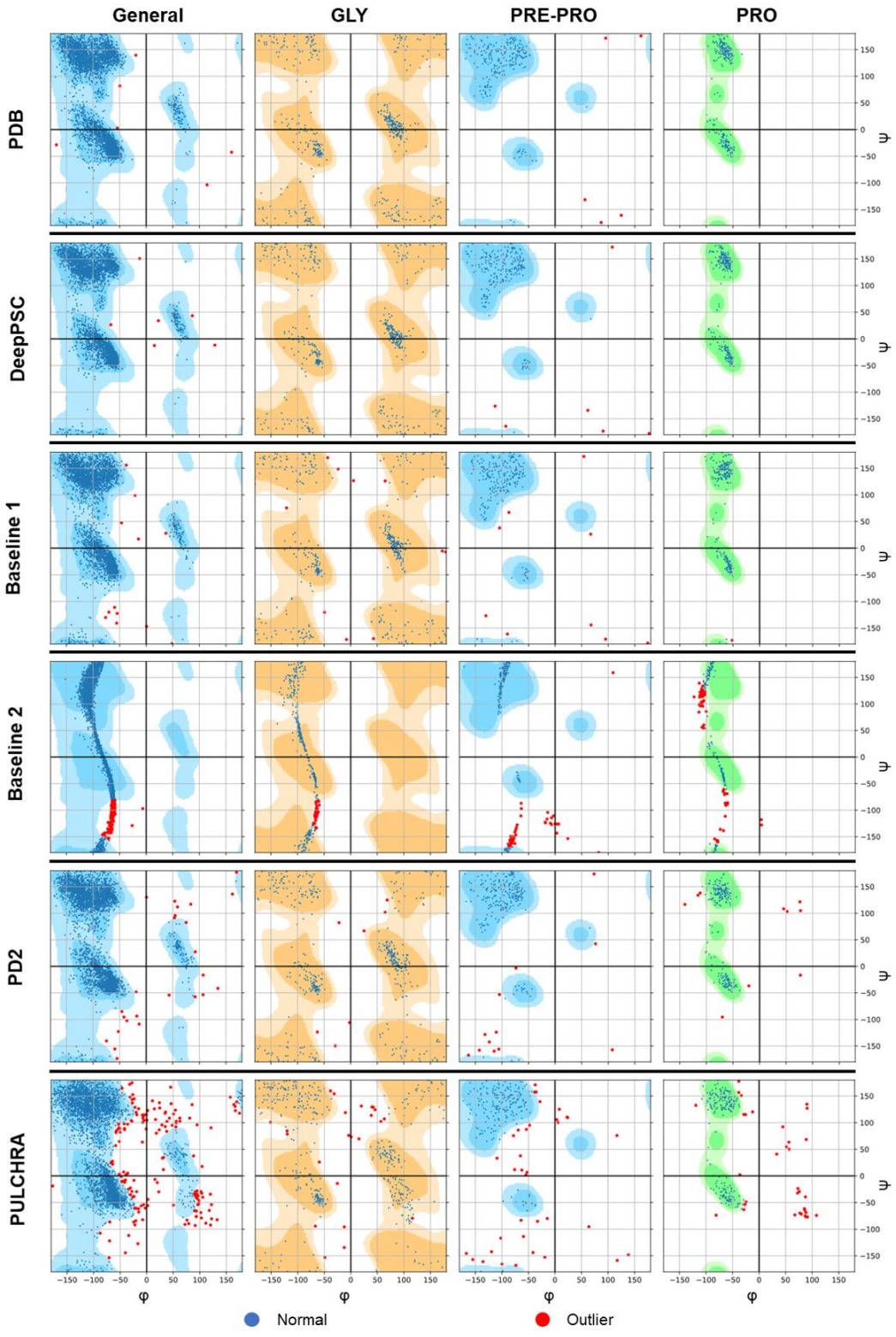


Figure 5. Ramachandran plots of the reconstructions obtained with different methods compared to the original structures (PDB). Rows represent the different methods and columns represent all residues (General), glycines (Gly), the residues preceding

prolines (Pre-Pro), and prolines (Pro). By taking the reference distributions in the backgrounds, residues are classified as normal residues (blue) or outliers (red).

The distributions of the atomic coordinate deviations were also used to calculate the GDT_P0.2 scores of DeepPSC, PD2, PULCHRA and the two baselines (Fig. 4), which clearly show that the backbones reconstructed by DeepPSC were more accurate than those obtained with PD2, PULCHRA and the two baselines. Lastly, the Ramachandran plots of the reconstructions obtained with different methods clearly showed that the backbone structures reconstructed by DeepPSC were the most reasonable among all methods (Fig. 5). In particular, none of the glycine and proline residues in the backbones obtained from DeepPSC were classified as outliers, consistent with the experimentally determined PDB structures, whereas many of these residues resulted as outliers in the backbones obtained by PD2 and PULCHRA, as well as by baseline 2. It is noteworthy that for baseline 2, the dihedral angles share a "S-shape" distribution pattern for all the four types of residues (general, glycine, pre-proline, and proline), which is consistent with the poor network fitting of this type of protein structure representation, as shown in Supplementary Fig. 6.

Conclusions

We consider protein structure representation as a critical problem in applying deep learning for reliable protein structure prediction, and for related endeavors such as

277 protein design. Our protein structure camera (PSC) approach provides a step forward
278 in protein structure representations, and toward enabling more sophisticated
279 applications of deep learning in biology.

280

281

282 **Methods**

283 **Geometrical calculation.** In this study, we represented C atoms and N atoms in peptide
284 planes as torsion angles by:

285 $$TC_n = \begin{cases} torsion\left(\overrightarrow{C\alpha_{n+1}C\alpha_{n+2}}, \overrightarrow{C\alpha_nC_n}, \overrightarrow{C\alpha_nC\alpha_{n+1}}\right), & n < L-2 \\ torsion\left(\overrightarrow{C\alpha_nC\alpha_{n-1}}, \overrightarrow{C\alpha_nC_n}, \overrightarrow{C\alpha_nC\alpha_{n+1}}\right), & n = L-1 \end{cases} \quad (1)$$

286 and

287 $$TN_n = \begin{cases} torsion\left(\overrightarrow{C\alpha_{n+1}C\alpha_{n+2}}, \overrightarrow{C\alpha_nN_n}, \overrightarrow{C\alpha_nC\alpha_{n+1}}\right), & n < L-2 \\ torsion\left(\overrightarrow{C\alpha_nC\alpha_{n-1}}, \overrightarrow{C\alpha_nN_n}, \overrightarrow{C\alpha_nC\alpha_{n+1}}\right), & n = L-1 \end{cases} \quad (2)$$

288 in which the torsion angle from $\overrightarrow{v_1}$ to $\overrightarrow{v_2}$ with $\vec{u}$ as axis was calculated by:

289 $$torsion(\overrightarrow{v_1}, \overrightarrow{v_2}, \vec{u}) = \arctan\left(\frac{n_1 \times n_2 \cdot \hat{u}}{n_1 \cdot n_2}\right), where \begin{cases} n_1 = \overrightarrow{v_1} \times \vec{u} \\ n_2 = \vec{u} \times \overrightarrow{v_2} \end{cases} \quad (3)$$

290 In the rebuilding process, the orientation of C_n and N_n was determined by rotating
291 the n^{th} peptide plane with $\overrightarrow{C\alpha_nC\alpha_{n+1}}$ as the axis:

292 $$\overrightarrow{PC_nC_n} = \begin{cases} rotation\left(\overrightarrow{PC\alpha_{n+2}C\alpha_{n+2}}, \widehat{C\alpha_nC\alpha_{n+1}}, TC_n\right), & n < L-2 \\ rotation\left(\overrightarrow{PC\alpha_{n-1}C\alpha_{n-1}}, \widehat{C\alpha_nC\alpha_{n+1}}, TC_n\right), & n = L-1 \end{cases} \quad (4)$$

293 and

294 $$\overrightarrow{PN_nN_n} = \begin{cases} rotation\left(\overrightarrow{PC\alpha_{n+2}C\alpha_{n+2}}, \widehat{C\alpha_nC\alpha_{n+1}}, TN_n\right), & n < L-2 \\ rotation\left(\overrightarrow{PC\alpha_{n-1}C\alpha_{n-1}}, \widehat{C\alpha_nC\alpha_{n+1}}, TN_n\right), & n = L-1 \end{cases} \quad (5)$$

295 where $PC\alpha_{n+2}$ and $PC\alpha_{n-1}$ is the projections of $C\alpha_{n+2}$ and $C\alpha_{n-1}$ on
296 $\overrightarrow{C\alpha_nC\alpha_{n+1}}$, respectively. The rotation was calculated by Rodrigues-Gibbs

Formulation[42]:

$$\overrightarrow{v_2} = rotation(\overrightarrow{v_1}, \hat{u}, T) = \overrightarrow{v_1}\cos T + \hat{u} \times \overrightarrow{v_1}\sin T + (\overrightarrow{v_1} \cdot \hat{u})\hat{u}(1 - \cos T) \quad (6)$$

in which $\overrightarrow{v_2}$ was obtained by rotate $\overrightarrow{v_1}$ with a unit vector $\hat{u}$ as axis and T as torsion angle. We assumed $\overrightarrow{PO_nO_n} \equiv \overrightarrow{PC_nC_n}$ using the ideal peptide plane conformation. Afterward the relative locations from atoms C_n, N_n, and O_n to $C\alpha_n$ were respectively determined by:

$$\begin{cases} \overrightarrow{C\alpha_nC_n} = C\alpha_n\widehat{C\alpha_{n+1}} * \left|\overrightarrow{C\alpha_nPC_n}\right| + P\widehat{C_nC_n} * \left|\overrightarrow{PC_nC_n}\right| \\ \overrightarrow{C\alpha_nO_n} = C\alpha_n\widehat{C\alpha_{n+1}} * \left|\overrightarrow{C\alpha_nPO_n}\right| + P\widehat{O_nO_n} * \left|\overrightarrow{PO_nO_n}\right| \\ \overrightarrow{C\alpha_nN_n} = C\alpha_n\widehat{C\alpha_{n+1}} * \left|\overrightarrow{C\alpha_nPN_n}\right| + P\widehat{N_nN_n} * \left|\overrightarrow{PN_nN_n}\right| \end{cases} \quad (7)$$

where $\left|\overrightarrow{C\alpha_nPC_n}\right|$, $\left|\overrightarrow{PC_nC_n}\right|$, $\left|\overrightarrow{C\alpha_nPO_n}\right|$, $\left|\overrightarrow{PO_nO_n}\right|$, $\left|\overrightarrow{C\alpha_nPN_n}\right|$, and $\left|\overrightarrow{PN_nN_n}\right|$ are a group of fixed length estimated from training data (listed in Table S1). Note that there are two type of peptide planes and the fixed lengths are correspondingly different. In the *trans* peptide plane, the distance between adjacent Cα is approximately 3.8 Å while for the *cis* peptide plane it is approximately 3.0 Å. Therefore, we rebuilt the n^{th} peptide plane with fixed lengths for the *trans* peptide plane when $\left|\overrightarrow{C\alpha_nC\alpha_{n+1}}\right| \geq 3.4$ Å, otherwise with fixed lengths for *cis* peptide plane.

Finally, the coordinates were determined as:

$$\begin{cases} C_n = C\alpha_n + \overrightarrow{C\alpha_nC_n} \\ O_n = C\alpha_n + \overrightarrow{C\alpha_nO_n} \\ N_n = C\alpha_n + \overrightarrow{C\alpha_nN_n} \end{cases} \quad (8)$$

The rebuilt atoms C_n, N_{n-1} and known $C\alpha_n$ in a residue were used to rebuild atom $C\beta_n \in \mathbb{R}^3$ with the following process. First, we initialized $\overrightarrow{C\alpha_nC\beta_n'}$ at the middle between $C\widehat{\alpha_nC_n}$ and $C\widehat{\alpha_nN_{n-1}}$ by:

$$\overrightarrow{C\alpha_nC\beta_n'} = \frac{C\widehat{\alpha_nC_n} + C\widehat{\alpha_nN_{n-1}}}{2} * \left|\overrightarrow{C\alpha_nC\beta_n}\right| \quad (9)$$

Afterward, $\overrightarrow{C\alpha_n C\beta_n}$ was determined by rotating $\overrightarrow{C\alpha_n C\beta_n'}$ with $\widehat{N_{n-1}C_n}$ as axis and $TC\beta$ as torsion angle:

$$\overrightarrow{C\alpha_n C\beta_n} = \text{rotation}\left(\overrightarrow{C\alpha_n C\beta_n'},\ \widehat{N_{n-1}C_n}, TC\beta\right) \tag{10}$$

We then calculated $C\beta_n$ by:

$$C\beta_n = C\alpha_n + \overrightarrow{C\alpha_n C\beta_n} \tag{11}$$

The fixed length $\left|\overrightarrow{C\alpha_n C\beta_n}\right|$ and fixed angle $TC\beta$ used above are constants estimated from the training data (listed in Supplementary Table 1).

Dataset. We selected a subgroup from the protein structures reported in the PDB, to build a tertiary structure dataset. A PDB entry was not included in this subgroup if: (i) the structure was not determined by X-ray crystallography; (ii) the entry has a number of residues fewer than 15 or higher than 800; (iii) the entry has missing atoms in the backbone or unnatural residues; (iv) the entry has sequence identity higher than 40% with another entry included in the subgroup[30]. This resulted in the construction of a non-redundant dataset containing 10,302 protein structures.

Model training. We used 10-fold cross validation in training our models. The whole dataset was randomly and equally separated into ten sub-datasets. We routinely used one sub-dataset as the validation set and all the other nine sub-datasets as the model training sets. All the models were trained for 30 epochs using mean square error (MSE) as the loss function and the Adam optimizer[43]. The training batch size for the local structure embedding block was the total number of residues of the input structures, thus

it was dynamic even if the number of input structures was fixed. To maintain the training batch relatively stable, we split all training structures as batches containing 1~5 structures with approximately 800 residues. The learning rate was set to 0.0003 for the first 3 epochs, and then was adjusted according to the cosine-annealing schedule[44] in the following epochs. The trained models were validated with the corresponding validation set after every epoch. The curves of validation loss showed that all the models of DeepPSC and baseline 1 steadily converged at the 30th epoch, while the models of baseline 2 showed poor fitting (Supplementary Fig. 6). The network construction and model training were implemented with PyTorch, an open source machine learning framework. All the details we have not mention follow the default setting of PyTorch.

Performance criteria. In this study the performance of various methods was evaluated on the basis of three criteria, *e.g.*, root mean square deviation (RMSD)[45], global distance test (GDT)[46], and Ramachandran (RAMA) outliers[47]. RMSD is one of the most used criteria to measure the similarity between two structures [45], and is calculated as follows:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N} d_i}{N}} \tag{12}$$

where d_i is the coordinate deviation between atom i in two structures, and N is the total number of atoms. Considering that RMSD usually increases as the number of atoms of a protein increases[48], this value is usually normalized as $RMSD_{100}$, which describes the same deviation in 100 atoms[49], and is calculated as follows:

$$RMSD_{100} = \frac{RMSD}{1 + \ln\sqrt{\frac{N}{100}}} \quad (13)$$

where N is the number of atoms.

RMSD is, however, strongly affected by the parts in the structures that deviate the most, therefore it often fails to represent the deviations of most of the atoms. Aimed at alleviating this problem, a community-wide experiment called CASP (critical assessment of techniques for Protein Structure Prediction)[46] have been using a different indicator, the Global Distance Test (GDT), as their main assessment method for ranking protein structure prediction methods. GDT scores are calculated as the percentage of atoms that have distance deviations smaller than the preset distance cutoffs. Cutoffs for GDT in CASP is usually set to 1, 2, 4, and 8 Å. In this study, the cutoff was set to 0.2 Å, and the GDT was labeled as GDT_P0.2.

The Ramachandran Plot is a statistical reference distribution of the combination of the backbone dihedral angles in proteins[47]. In a Ramachandran Plot, one can classify residues in a given protein backbone structure as ‘core’, ‘allowed’, and ‘outliers’. The percentage of outliers (RAMA outliers) is used to assess protein backbone structure uncertainty.

Acknowledgements

This work was supported by the National Key R&D Program of China (2018YFA0901000), and the Guangzhou Science and Technology Program key projects (201904020016). Dr. Hongmin Cai acknowledges the support by the Key-Area Research and Development of Guangdong Province under Grant (2020B010166002, 2020B1111190001), the National Natural Science Foundation of China (61771007), the Health & Medical Collaborative Innovation Project of Guangzhou City (201803010021, 202002020049).

Author contributions

X.Z. contributed to the experimental design, methodology, coding, data analysis and writing of the original draft. J.L. contributed to the methodology, coding and data analysis. Y.C contributed to the data analysis. X.Y. contributed to the data analysis and writing (review and revision). W.Z. contributed to the data analysis. H.C. contributed to the methodology, data analysis and writing (review and revision), and was partially responsible for supervision. Z.L. was responsible for the experimental design, supervision as well as funding acquisition, and contributed to the data analysis, writing (review and revision). All authors reviewed and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Code and data availability

All source codes and models of DeepPSC are openly available on GitHub (https://github.com/EricZhangSCUT/DeepPSC), together with the PDB ID lists of all involved datasets.

404 **Reference**

405 1 Smyth, M. S. & Martin, J. H. X-ray crystallography. *Mol. Pathol.* **53**, 8-14
406 (2000).

407 2 Voula Kanelis, J. D. F.-K. & Lewis E. Kay. Multidimensional NMR Methods
408 for Protein Structure Determination. *IUBMB Life* **52**, 291-302 (2001).

409 3 Cheng, Y. Single-particle cryo-EM - How did it get here and where will it go.
410 *Science* **361**, 876-880 (2018).

411 4 Senior, A. W. *et al.* Improved protein structure prediction using potentials from
412 deep learning. *Nature* **577**, 706-710 (2020).

413 5 AlQuraishi, M. End-to-end differentiable learning of protein structure. *Cell Syst.*
414 **8**, 292-301 e293 (2019).

415 6 Wang, S., Sun, S., Li, Z., Zhang, R. & Xu, J. Accurate de novo prediction of
416 protein contact map by ultra-deep learning model. *PLoS Comput. Biol.* **13**,
417 e1005324 (2017).

418 7 Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified
419 rational protein engineering with sequence-based deep representation learning.
420 *Nat. Methods* **16**, 1315-1322 (2019).

421 8 Rao R, B. N., Thomas N, et al. Evaluating protein transfer learning with TAPE.
422 Preprint at https://arxiv.org/abs/1906.08230 (2019).

423 9 Hou, J., Adhikari, B. & Cheng, J. DeepSF: deep convolutional neural network
424 for mapping protein sequences to folds. *Bioinformatics* **34**, 1295-1303 (2018).

425 10 Wang, S., Peng, J., Ma, J. & Xu, J. Protein secondary structure prediction using

426 deep convolutional neural fields. *Sci. Rep.* **6**, 18962 (2016).

427 11 Kulmanov, M., Khan, M. A., Hoehndorf, R. & Wren, J. DeepGO: predicting
428 protein functions from sequence and interactions using a deep ontology-aware
429 classifier. *Bioinformatics* **34**, 660-668 (2018).

430 12 Tsubaki, M., Tomii, K. & Sese, J. Compound-protein interaction prediction with
431 end-to-end learning of neural networks for graphs and sequences.
432 *Bioinformatics* **35**, 309-318 (2019).

433 13 Wang, J., Cao, H., Zhang, J. Z. H. & Qi, Y. Computational protein design with
434 deep learning neural networks. *Sci. Rep.* **8**, 6349 (2018).

435 14 Zheng, S., Li, Y., Chen, S., Xu, J. & Yang, Y. Predicting drug–protein interaction
436 using quasi-visual question answering system. *Nat. Mach. Intell.* **2**, 134-140
437 (2020).

438 15 Stepniewska-Dziubinska, M. M., Zielenkiewicz, P. & Siedlecki, P.
439 Development and evaluation of a deep learning model for protein-ligand
440 binding affinity prediction. *Bioinformatics* **34**, 3666-3674 (2018).

441 16 Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep
442 convolutional neural networks. *NeurIPS* 1097-1105 (2012).

443 17 Zhao, Z.Q., Zheng, P., Xu, S.t. & Wu, X. Object detection with deep learning:
444 A review. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 3212-3232 (2019).

445 18 Sun, Y., Chen, Y., Wang, X. & Tang, X. Deep learning face representation by
446 joint identification-verification. *NeurIPS* 1988-1996 (2014).

447 19 Mallat, S. Understanding deep convolutional networks. *Philos. Trans. A Math.*

Phys. Eng. Sci. **374**, 20150203 (2016).

20 Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. *European conference on computer vision* 818-833 (2014).

21 Adams, P. D. *et al.* PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 213-221 (2010).

22 Emsley, P. & Cowtan, K. Coot: model-building tools for molecular graphics. *Acta Crystallogr. D Biol. Crystallogr.* **60**, 2126-2132 (2004).

23 Rotkiewicz, P. & Skolnick, J. Fast procedure for reconstruction of full-atom protein models from reduced representations. *J. Comput. Chem.* **29**, 1460-1465 (2008).

24 Esnouf, R. M. Polyalanine Reconstruction from Ca Positions Using the Program CALPHA Can Aid Initial Phasing of Data by Molecular Replacement Procedures. *Acta Crystallogr. D Biol. Crystallogr.* **53**, 665-672 (1997).

25 Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. Features and development of Coot. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 486-501 (2010).

26 Roy, A., Kucukural, A. & Zhang, Y. I-TASSER: a unified platform for automated protein structure and function prediction. *Nat. Protoc.* **5**, 725-738 (2010).

27 Li, Y. & Zhang, Y. REMO: A new protocol to refine full atomic protein models from C-alpha traces by optimizing hydrogen-bonding networks. *Proteins* **76**, 665-676 (2009).

28 Gront, D., Kmiecik, S. & Kolinski, A. Backbone building from quadrilaterals: a fast and accurate algorithm for protein backbone reconstruction from alpha carbon coordinates. *J. Comput. Chem.* **28**, 1593-1597 (2007).

29 Maupetit, J., Gautier, R. & Tuffery, P. SABBAC: online Structural Alphabet-based protein BackBone reconstruction from Alpha-Carbon trace. *Nucleic Acids Res.* **34**, W147-151 (2006).

30 Moore, B. L., Kelley, L. A., Barber, J., Murray, J. W. & MacDonald, J. T. High-quality protein backbone reconstruction from alpha carbons using Gaussian mixture models. *J. Comput. Chem.* **34**, 1881-1889 (2013).

31 Zhang, R. *et al.* 4.4 A cryo-EM structure of an enveloped alphavirus Venezuelan equine encephalitis virus. *EMBO J.* **30**, 3854-3863 (2011).

32 Xiong, D., Zeng, J. & Gong, H. A deep learning framework for improving long-range residue-residue contact prediction using a hierarchical strategy. *Bioinformatics* **33**, 2675-2683 (2017).

33 Kyte, J. & Russell F. Doolittle. A simple method for displaying the hydropathic character of a protein. *J. of Mol. Biol.* **157**, 105-132 (1982).

34 Zimmerman, J. M., Naomi Eliezer & R. Simha. The characterization of amino acid sequences in proteins by statistical methods. *J. of Theor. Biol.* **21.2**, 170-201 (1968).

35 Huang, F. & Nau, W. M. A conformational flexibility scale for amino acids in peptides. *Angew. Chem. Int. Ed. Engl.* **42**, 2269-2272 (2003).

36 Payne, P. W. Reconstruction of protein conformations from estimated positions

of the Cα coordinates. *Protein Sci.* **2(3)**, 315-324 (1993).

37 He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 770-778 (2016).

38 Graves, A. & Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **18**, 602-610 (2005).

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

40 Rumelhart, D. E., Hinton, G. E. & Williams, R. J. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1* 318-362 (MIT Press, Cambridge, 1986).

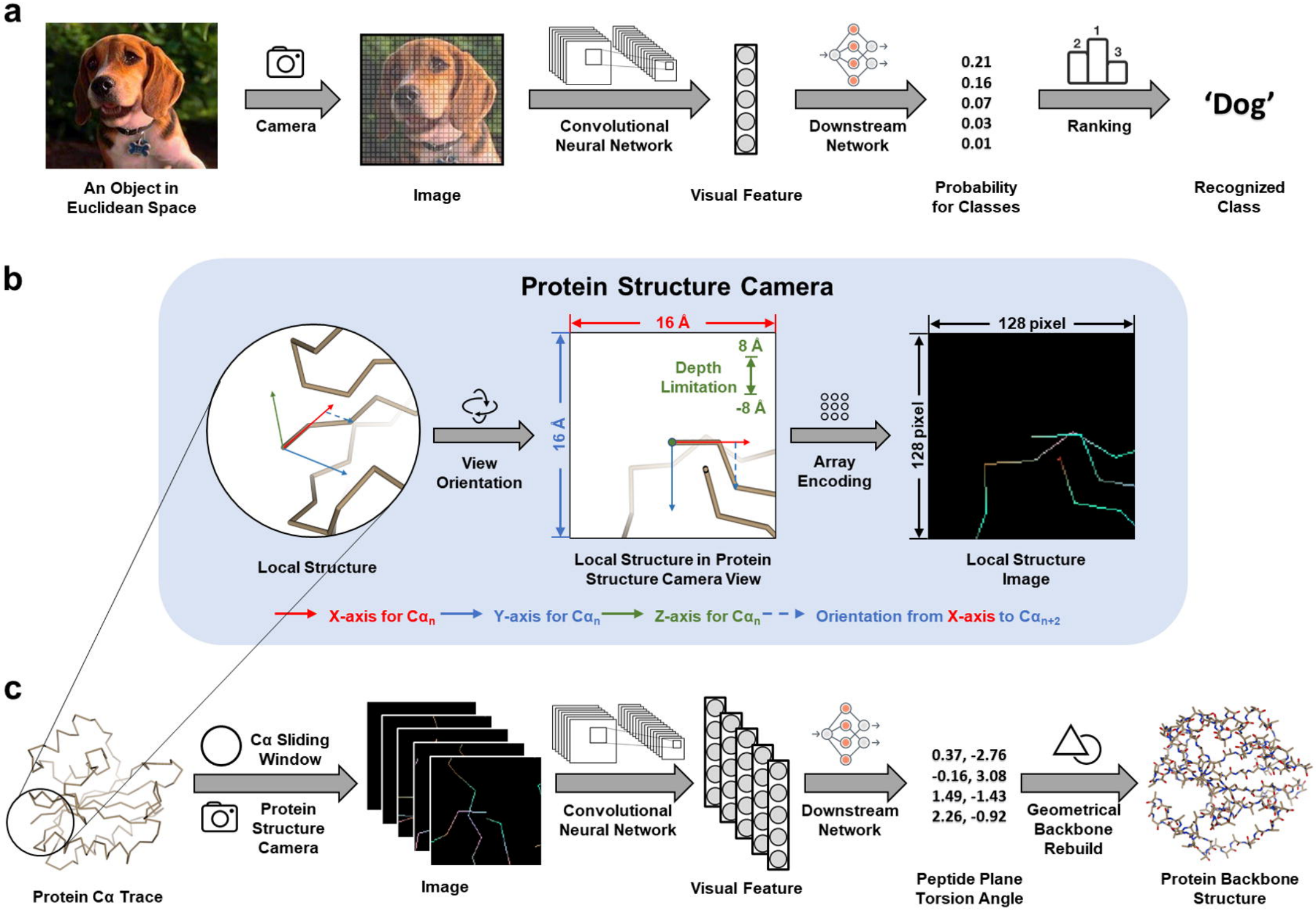
41 Scapin, G., Potter, C. S. & Carragher, B. Cryo-EM for small molecules discovery, design, understanding, and application. *Cell Chem. Biol.* **25**, 1318-1325 (2018).

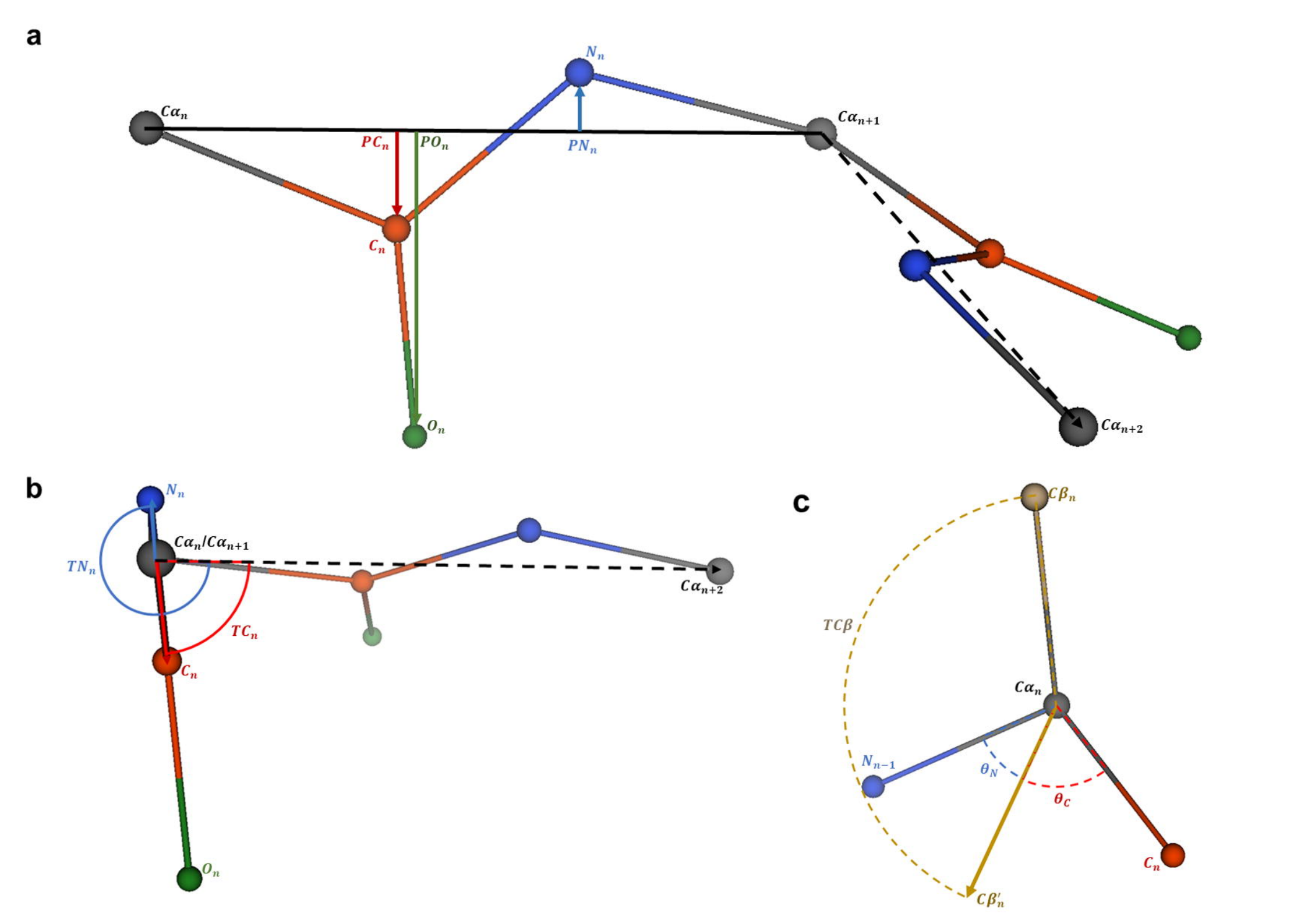
42 Parsons, J., Holmes, J. B., Rojas, J. M., Tsai, J. & Strauss, C. E. Practical conversion from torsion space to Cartesian space for in silico protein synthesis. *J. Comput. Chem.* **26**, 1063-1068 (2005).

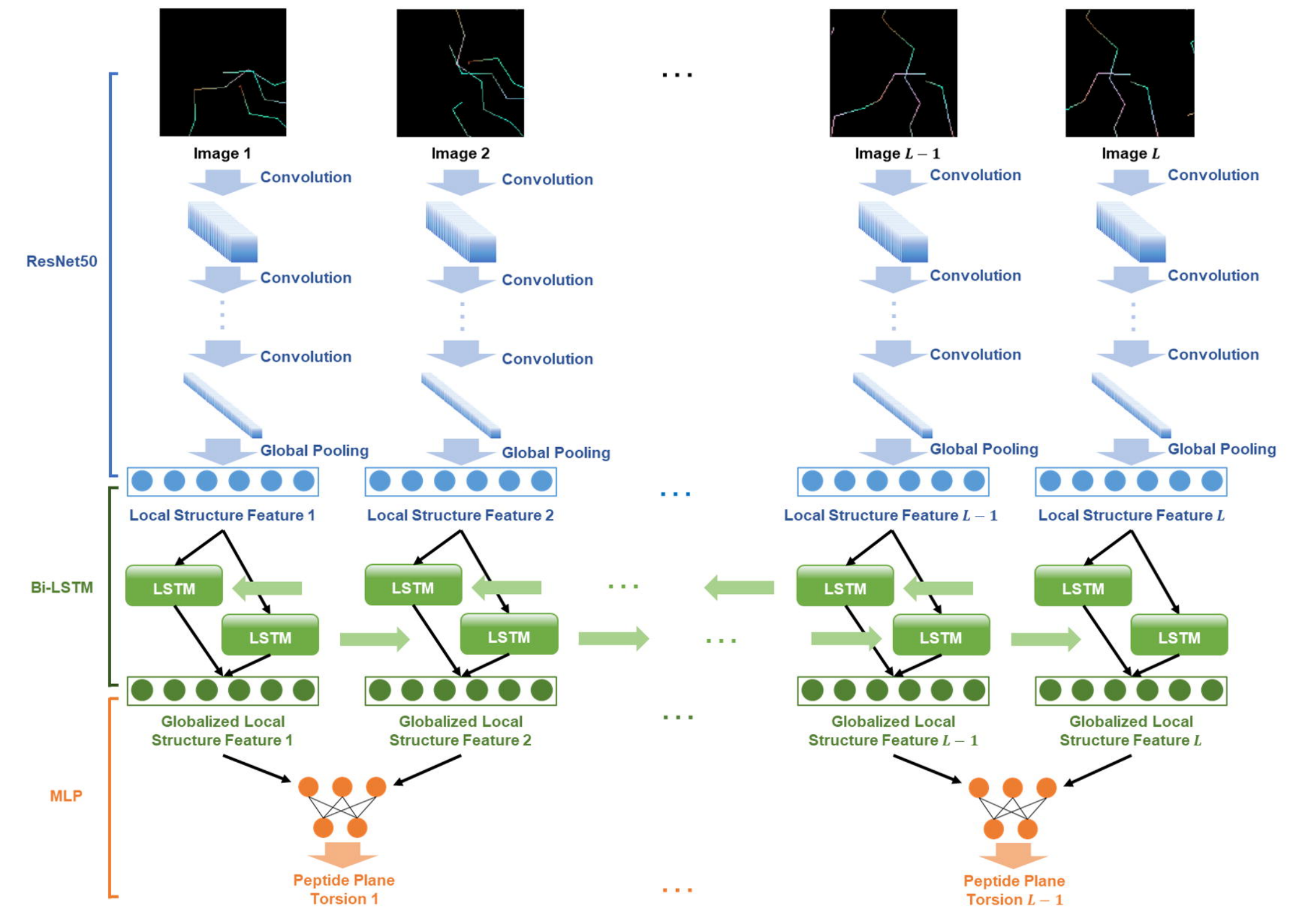
43 Kingma, D. P. & Jimmy Ba. Adam: A method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).

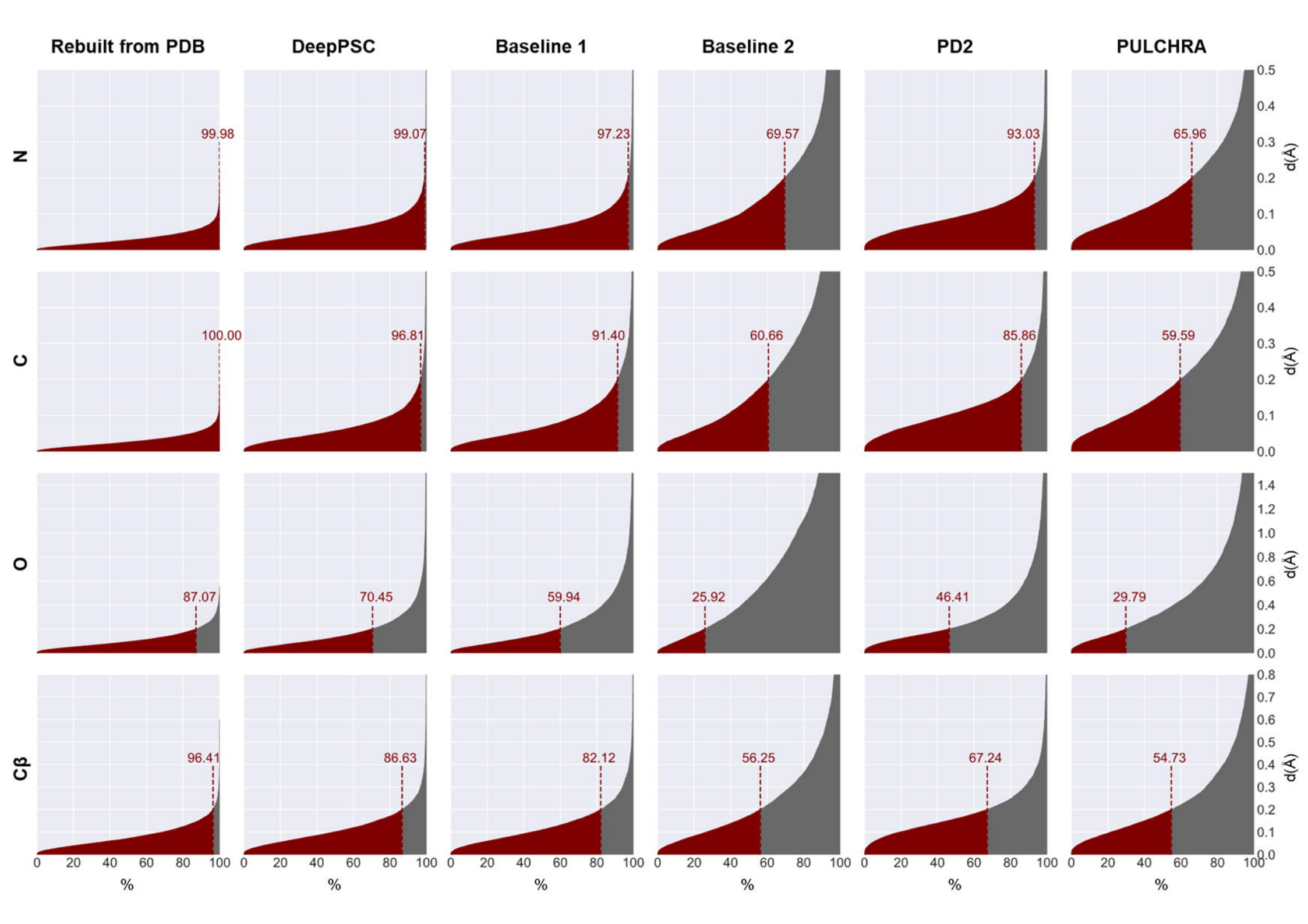
44 Loshchilov, I. & Frank Hutter. SGDR: Stochastic Gradient Descent with warm Restarts. Preprint at https://arxiv.org/abs/1608.03983 (2016).

514 45 Kufareva, I. & Abagyan, R. Methods of protein structure comparison. *Methods*
515 *Mol. Biol.* **857**, 231-257 (2012).
516 46 Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T. & Tramontano, A.
517 Critical assessment of methods of protein structure prediction (CASP) - round
518 x. *Proteins* **82 Suppl 2**, 1-6 (2014).
519 47 Lovell, S. C. *et al.* Structure validation by Cα geometry: ϕ,ψ and Cβ deviation.
520 *Proteins* **50**, 437-450 (2003).
521 48 Sargsyan, K., Grauffel, C. & Lim, C. How molecular size impacts RMSD
522 applications in molecular dynamics simulations. *J. Chem. Theory Comput.* **13**,
523 1518-1524 (2017).
524 49 Carugo, O. & Pongor, S. A normalized root-mean-square distance for
525 comparing protein three-dimensional structures. *Protein Sci.* **10**, 1470-1473
526 (2001).
527 50 Pettersen, E. F. *et al.* UCSF Chimera - A visualization system for exploratory
528 research and analysis. *J. Comput. Chem.* **25**, 1605-1612 (2004).
529


a
Camera
Convolutional Neural Network
Downstream Network
0.21
0.16
0.07
0.03
0.01
Ranking
'Dog'
An Object in Euclidean Space
Image
Visual Feature
Probability for Classes
Recognized Class
b
Protein Structure Camera
16 Å
8 Å
Depth Limitation
-8 Å
16 Å
128 pixel
128 pixel
View Orientation
Array Encoding
Local Structure
Local Structure in Protein Structure Camera View
Local Structure Image
X-axis for Cα$_n$
Y-axis for Cα$_n$
Z-axis for Cα$_n$
Orientation from X-axis to Cα$_{n+2}$
c
Cα Sliding Window
Protein Structure Camera
Convolutional Neural Network
Downstream Network
0.37, -2.76
-0.16, 3.08
1.49, -1.43
2.26, -0.92
Geometrical Backbone Rebuild
Protein Cα Trace
Image
Visual Feature
Peptide Plane Torsion Angle
Protein Backbone Structure


a
N_n
$C\alpha_n$
$C\alpha_{n+1}$
PC_n
PO_n
PN_n
C_n
O_n
$C\alpha_{n+2}$
b
N_n
$C\alpha_n/C\alpha_{n+1}$
TN_n
TC_n
C_n
$C\alpha_{n+2}$
O_n
c
$C\beta_n$
$TC\beta$
$C\alpha_n$
N_{n-1}
θ_N
θ_C
C_n
$C\beta'_n$


ResNet50
Bi-LSTM
MLP
Image 1
Image 2
Image $L-1$
Image L
Convolution
Global Pooling
Local Structure Feature 1
Local Structure Feature 2
Local Structure Feature $L-1$
Local Structure Feature L
LSTM
Globalized Local Structure Feature 1
Globalized Local Structure Feature 2
Globalized Local Structure Feature $L-1$
Globalized Local Structure Feature L
Peptide Plane Torsion 1
Peptide Plane Torsion $L-1$


Rebuilt from PDB
DeepPSC
Baseline 1
Baseline 2
PD2
PULCHRA
N
C
O
Cβ
99.98
99.07
97.23
69.57
93.03
65.96
100.00
96.81
91.40
60.66
85.86
59.59
87.07
70.45
59.94
25.92
46.41
29.79
96.41
86.63
82.12
56.25
67.24
54.73
d(Å)
%
0
20
40
60
80
100