nature structural & molecular biology

Supplementary information

https://doi.org/10.1038/s41594-025-01713-3

CryoAtom improves model building for cryo-EM

In the format provided by the authors and unedited

Supplementary Information

S1. Performance evaluation

S1.1 Running time and GPU memory usage

We evaluated CryoAtom's running time as a function of protein length on the 177 high-resolution maps using a single GPU (A100, ~15G memory) (Fig. S1a). The running time increases linearly with protein length. CryoAtom can build a structure of approximately 40,000 residues in about 3.5 hours. CryoAtom allows users to balance GPU memory and running time. By default, CryoAtom processes 300 residues at a time, using ~13 GB of GPU memory. An additional ~1 GB of memory is required for every 100 extra residues. Fig. S1b illustrates the running time for the PDB entry 8FNV (>9,000 residues) under three configurations. Increasing GPU memory by 1.6 times (from 15GB to 24GB) results in acceleration of the speed by 1.5 times (from 44 minutes to 30 minutes). This linear relationship can be generalized to proteins of any length. In additional tests, the accuracy of predicted models remains consistent with increased crop size.

S1.2 Evaluation metrics

We used the same set of evaluation metrics defined by ModelAngelo [1], including *backbone recall*, *backbone precision*, *backbone RMSD*, *Cα RMSD*, *amino acid accuracy*, and *completeness*.

- *Backbone recall* is the fraction of deposited residues (represented by Cα atoms) that have a predicted residue (represented by Cα atom) within 3 Å.
- *Backbone precision* is the fraction of predicted residues (represented by Cα atoms) that have a deposited residue (represented by Cα atom) within 3 Å.

The remaining subsequent metrics involve the matching between deposited residues and predicted residues, and only consider the deposited residues that have a predicted residue within 3 Å. We first calculate the distances between the Cα atoms in predicted structure and the Cα atoms in deposited structure. The Hungarian Matching Algorithm[2] is then used to match the predicted residues with the deposited residues such that the total distance is minimized. The metrics can be defined based on the matched residue pairs.

Let *p* represent the number of matched residues that have the same amino acid identity; *m* represent the total number of matched residues; and *n* represent the total number of residues in the deposited structure.

- *Cα RMSD* is the root-mean-square deviation (RMSD) between the Cα atoms of the matched residue pairs.
- *Backbone RMSD* is similar to the former but includes all four main-chain atoms (Cα, C, O, and N).
- *Amino acid accuracy* is the fraction of residue pairs that share identical amino acid types (i.e., *p/m*) .
- *Completeness* is the fraction of all deposited residues (including those unmatched residues) that have a matched predicted residue with the same amino acid identity (i.e., *p/n*). It is worth noting that *Completeness* ≈ *Backbone recall* × *Amino acid accuracy.*

S1.3 Sensitivity analysis of CryoAtom to hyperparameters

CryoAtom has many hyperparameters during the inference phase. Here, we analyze two key hyperparameters: the prediction threshold for the classification network in Stage 1 (denoted as *t*) and the number of recycling rounds in Stage 2 (denoted as *n*). In the default settings of CryoAtom, *t* is set to 0.6 and *n* to 3. We performed tests to evaluate the effects of varying these parameters.

As illustrated in Fig. S3a, reducing *t* and *n* affects the completeness, with *n* having a stronger impact on the completeness. This suggests that recycling effectively enhances the model completeness. Additionally, when lowering the Stage 1 threshold to $t = 0.4$, the backbone precision remains largely unaffected (see Fig. S3b), indicating that CryoAtom effectively filters out false positives generated in Stage 1.

We also tested the impact of randomly rotating and translating density maps. We fed the transformed density map into CryoAtom. Then, the models output by CryoAtom are aligned with the native models using US-align [3] for comparison. The results were interesting: appropriate rotation and translation can improve model quality (see Fig. S3), although models built from the original and the transformed maps remain broadly similar. These data demonstrate the robustness of CryoAtom.

S1.4 Map masking improves backbone precision

One factor contributing to CryoAtom's lower precision is the absence of automatic map masking. Map masking is typically used to exclude regions of low resolution or those unrelated to the structures of interest (e.g., membrane or solvent). We evaluated the impact of map masking on a set of 70 maps from the 177 high-resolution maps, for which both the original and masked maps are available from EMDB [4]. The average backbone precision for structures built by CryoAtom increased from 86.0% (using the original map) to 90.7% (using the masked map). While most cases show no significant difference between the original and the masked maps (Fig. S5), some examples demonstrate improved backbone precision with masked maps. For instance, for the map EMD-33306, the backbone precision improved from 25.2% to 97.1% after masking (Fig. S5e). However, map masking can sometimes result in incomplete atomic structures (Fig. S5f). Nonetheless, our package provides an optional argument to accept masked maps for expert users.

S1.5 Evaluation of residue confidence scores

CryoAtom provides a reliable estimation of model accuracy through a per-residue confidence score, similar to the confidence scores available in protein structure prediction and ModelAngelo (Box 1). This score is derived from the predicted FAPE loss (see Methods). The confidence score ranges from 0 to 100; a higher score indicating lower FAPE loss and a more confident prediction. This score is stored in the B-factor field of the mmCIF file. We analyzed the correlation between the confidence scores of all residues in 177 predicted structures generated by CryoAtom and their backbone RMSDs (Fig. S6a). Generally, a higher residue confidence score correlates with a lower backbone RMSD. In Fig. S6b-c, we illustrate two examples colored by their respective confidence scores. Notably, CryoAtom assigns lower confidence scores to loop regions, which tend to be more flexible and exhibit lower resolution in density maps.

S1.6 Performance on a non-redundant set of maps

To investigate the impact of redundancy between training and test maps, the 281 test maps (from both the high-resolution and the low-resolution test sets) are compared with the training maps at a 40% sequence identity threshold and a TM-score threshold of 0.5, resulting in a non-redundant set of 54 maps (43 and 11 from the high-resolution and the low-resolution test sets, respectively).

The performance of CryoAtom is listed in Table S4. Surprisingly, the CryoAtom models for the non-redundant maps are more complete than the redundant maps (completeness 74% vs. 66%). By dividing these maps into two subsets according to their resolutions, we can see that redundancy has less impact on the high-resolution maps (completeness 82% vs. 84%). However, the model completeness for the non-redundant low-resolution maps is even higher than the full test set (41% vs. 37%). This is because, before removing redundancy, there were 32 poor models in the low-resolution dataset, where the completeness of the CryoAtom models were below 20%. Only three such models were retained in the non-redundant set, resulting in higher completeness. In summary, these results indicate that the key factor impacting CryoAtom's performance is the map resolution rather than the sequence similarity or structural similarity to the training data.

Overall, CryoAtom's performance on the non-redundant dataset is generally consistent with previous observations (see Fig. 2, 4, and Fig. S8). Except for the backbone precision, which is lower than that of ModelAngelo (95.5% vs. 99.2%), CryoAtom outperforms ModelAngelo in all other metrics (completeness 73.6% vs. 65.5%, backbone RMSD 0.47 Å vs. 0.58 Å, backbone recall 88.7% vs. 81.0%). In terms of model quality assessment, although CryoAtom predicted more residues in the more challenging low-resolution region, it still achieved competitive scores compared to ModelAngelo (MolProbity scores 3.65 vs. 3.75, EMRinger 2.34 vs. 2.35). Finally, we compare CryoAtom and ModelAngelo on the intermediate models (model_net) without any post-processing. The results (see Fig. S8f) show that CryoAtom has a higher backbone recall (95.8% vs. 89.7%) and higher amino acid accuracy (65.2% vs. 50.5%). This demonstrates the superiority of the Cryo-Net in constructing atomic positions and recognizing amino acid types.

S2. Network details

S2.1 U-Net

The encoder of the U-Net [5] architecture adopts a bottleneck structure [6], which is the same as the convolutional neural network (CNN) module of Cryo-Net. The decoder uses the Res2Net architecture [7, 8], as shown in Fig. S11. The use of bottleneck for downsampling aims to preserve as much spatial information as possible, while the use of Res2Net for upsampling aims to extract as much semantic information as possible. The spatial information and semantic information are then combined to predict the probability map of Cα atoms.

S2.2 U-Net training

As mentioned in S2.1, the encoder architecture of U-Net is the same as the CNN network structure in Cryo-Net. In this case, we employed a form of transfer learning technique, where we used the pre-trained CNN network parameters from the Cryo-Net as the initial weights for the encoder of U-Net, without increasing the training sample size. This is because the Cryo-Net was thoroughly trained on various protein-related knowledge, such as structure prediction, amino acid classification, and confidence prediction. Therefore, the CNN network from Cryo-Net can extract richer semantic information from the cryo-EM density maps, which cannot be solely obtained through the task of predicting Cα positions in Stage 1. The encoder architecture of U-Net can extract richer information through this pre-training process.

S2.3 Sequence attention and IPA

The Sequence Attention and IPA modules are only briefly mentioned in the main text. In this section, they will be presented in the form of pseudocodes.

Algorithm S1: Sequence attention

def `Seq-attention`$(\{\boldsymbol{s}_i\}, \{\boldsymbol{e}_j\}, N_{head} = 8, c = 48)$**:**

`// Input projections`

$\boldsymbol{s}_i \leftarrow \mathrm{LayerNorm}(\boldsymbol{s}_i)$;

$\boldsymbol{q}_i^h = \mathrm{LinearNoBias}(\boldsymbol{s}_i)$ $\quad \boldsymbol{q}_i^h \in \mathbb{R}^c, h \in \{1, ..., N_{head}\}$;

$\boldsymbol{k}_j^h, \boldsymbol{v}_j^h = \mathrm{LinearNoBias}(\boldsymbol{e}_j)$ $\quad \boldsymbol{k}_j^h, \boldsymbol{v}_j^h \in \mathbb{R}^c, h \in \{1, ..., N_{head}\}$;

$\boldsymbol{g}_i^h = \mathrm{sigmoid}(\mathrm{Linear}(\boldsymbol{s}_i))$ $\quad \boldsymbol{g}_i^h \in \mathbb{R}^c$;

`// Cross attention`

$a_{ij}^h = \mathrm{softmax}_j\left(\frac{1}{\sqrt{c}} \boldsymbol{q}_i^{h\top} \boldsymbol{k}_j^h\right)$;

$\boldsymbol{o}_i^h = \boldsymbol{g}_i^h \odot \sum_j a_{ij}^h \boldsymbol{v}_j^h$;

`// Output projections`

$\tilde{\boldsymbol{s}}_i = \mathrm{Linear}(\mathrm{concat}_h(\boldsymbol{o}_i^h))$;

return $\{\tilde{\boldsymbol{s}}_i\}$

In this algorithm, the input $\boldsymbol{s}_i$ represents the node representation of the i-th node, and the input $\boldsymbol{e}_j$ represents the ESM-2 [9] embedding representation of the j-th amino acid in the sequence. The output $\tilde{\boldsymbol{s}}_i$ is the updated node representation of the i-th node.

Algorithm S2: Local invariant point attention with 3D-RoPE

def IPA($\{\boldsymbol{s}_j\}, \{\boldsymbol{z}_{ij}\}, \{T_j\}, \{\boldsymbol{x}_j\}, N_{head} = 10, c = 48, N_{query\ points} = 4, N_{point\ values} = 8$)**:**

$\boldsymbol{q}_j^h, \boldsymbol{k}_j^h, \boldsymbol{v}_j^h = \text{LinearNoBias}(\boldsymbol{s}_j)$ $\quad \boldsymbol{q}_j^h, \boldsymbol{k}_j^h, \boldsymbol{v}_j^h \in \mathbb{R}^c, h \in \{1, ..., N_{head}\}$;

$\vec{\boldsymbol{q}}_j^{hp}, \vec{\boldsymbol{k}}_j^{hp} = \text{LinearNoBias}(\boldsymbol{s}_j)$ $\quad \vec{\boldsymbol{q}}_j^{hp}, \vec{\boldsymbol{k}}_j^{hp} \in \mathbb{R}^3, p \in \{1, ..., N_{\text{query points}}\}$;

$\vec{\boldsymbol{v}}_j^{hp} = \text{LinearNoBias}(\boldsymbol{s}_j)$ $\quad \vec{\boldsymbol{v}}_j^{hp} \in \mathbb{R}^3, p \in \{1, ..., N_{\text{point values}}\}$;

$b_{ij}^h = \text{LinearNoBias}(\boldsymbol{z}_{ij})$;

$w_C = 0.1\sqrt{\frac{2}{N_{\text{query points}}}}$;

$w_L = \sqrt{\frac{1}{3}}$;

$\tilde{\boldsymbol{q}}_j^h, \tilde{\boldsymbol{k}}_j^h = \text{3D-RoPE}(\boldsymbol{q}_j^h, \boldsymbol{k}_j^h, \boldsymbol{x}_j)$;

$a_{ij}^h = \underset{1 \leqslant i \leqslant k}{softmax}\, w_L \left(\frac{1}{\sqrt{c}} \tilde{\boldsymbol{q}}_j^{h\top} \tilde{\boldsymbol{k}}_{j_i}^h + b_{ij}^h \right) - \frac{\gamma^h w_C}{2} \sum_p \left\| T_j \circ \vec{\boldsymbol{q}}_j^{hp} - T_{j_i} \circ \vec{\boldsymbol{k}}_{j_i}^{hp} \right\|^2$;

$\tilde{\boldsymbol{o}}_j^h = \sum_i a_{ij}^h \boldsymbol{z}_{ij}$;

$\boldsymbol{o}_j^h = \sum_i a_{ij}^h \boldsymbol{v}_{j_i}^h$;

$\vec{\boldsymbol{o}}_j^{hp} = T_j^{-1} \sum_i a_{ij}^h (T_{j_i} \circ \vec{\boldsymbol{v}}_{j_i}^{hp})$;

$\tilde{\boldsymbol{s}}_j = \text{Linear}(\text{concat}(\tilde{\boldsymbol{o}}_j^h, \boldsymbol{o}_j^h, \vec{\boldsymbol{o}}_j^{hp}, \left\| \vec{\boldsymbol{o}}_j^{hp} \right\|))$;

return $\{\tilde{\boldsymbol{s}}_j\}$

In this algorithm, the input $\boldsymbol{s}_j$ represents the node representation of the j-th node, $\boldsymbol{z}_{ij}$ represents the edge representation between the j-th node and its i-th neighbor, T_j represents the backbone frame of the j-th node, and $\boldsymbol{x}_j$ represents the position (i.e., the coordinates of the Cα atom) of the j-th node. The output $\tilde{\boldsymbol{s}}_j$ is the updated node representation of the j-th node. The highlighted text here outlines the main differences between the IPA module in this work and the one in AF2 [10]. The attention used here is a local form and includes a 3D rotary position embedding (see below).

S2.4 3D Rotary Position Embedding (3D-RoPE)

3D-RoPE is used in the node attention and IPA modules of the Cryo-Net. To describe this in detail, let's first introduce the 1D rotary position encoding [11]:

$$R_m = \begin{pmatrix} \cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \dots & 0 & 0 \\ \sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \dots & 0 & 0 \\ 0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\ 0 & 0 & 0 & 0 & \dots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} \end{pmatrix} \tag{S1}$$

$$\tilde{q}_m^T \tilde{k}_n = (R_m q_m)^T (R_n k_n) = q_m^T (R_m^T R_n) k_n = q_m^T R_{m-n} k_n \tag{S2}$$

where q and k represent the query vector and the key vector used in the attention mechanism, respectively. They are both d-dimensional vectors. The above formula is to introduce the positions m and n of the sequence into the attention mechanism using the multiplicative position encoding. We extend this process to 3D space:

$$R_{(x,y,z)} = \begin{pmatrix} R_x & 0 & 0 \\ 0 & R_y & 0 \\ 0 & 0 & R_z \end{pmatrix} \tag{S3}$$

$$\tilde{q}_m^T \tilde{k}_n = (R_{(x_m,y_m,z_m)} q_m)^T (R_{(x_n,y_n,z_n)} k_n) = q_m^T R_{(x_m-x_n,y_m-y_n,z_m-z_n)} k_n \tag{S4}$$

Here, we assume that the dimensions of the query and key vectors are both multiples of three, and (x, y, z) represents the position of each node (i.e., the coordinates of the Cα atoms). The 3D position encoding has richer connotations compared to the 1D position encoding, as it not only considers the distance information between nodes but also the directional information between nodes. Furthermore, this attention mechanism can actually be applied along the three coordinate axes:

$$q_m = \begin{pmatrix} q_m^x \\ q_m^y \\ q_m^z \end{pmatrix}, k_n = \begin{pmatrix} k_n^x \\ k_n^y \\ k_n^z \end{pmatrix} \tag{S5}$$

$$\begin{aligned} \tilde{q}_m^T \tilde{k}_n &= q_m^T R_{(x_m - x_n, y_m - y_n, z_m - z_n)} k_n \\ &= q_m^{xT} R_{x_m - x_n} k_n^x + q_m^{yT} R_{y_m - y_n} k_n^y + q_m^{zT} R_{z_m - z_n} k_n^z \end{aligned} \tag{S6}$$

The first one-third of the dimensions in the query and key vectors represent the information along the x-axis, the middle one-third represent the information along the y-axis, and the last one-third represent the information along the z-axis. From this perspective, the 3D-RoPE attention can be seen as the sum of three 1D-RoPE attentions in different directions (x, y, z). The 3D-RoPE attention is realized by the inner product calculation. In the 3D world, the inner product has a clear physical meaning, such as representing the work done:

$$\begin{aligned} W &= \vec{F} \bullet \vec{s} = (\vec{F_x} + \vec{F_y} + \vec{F_z}) \bullet (\vec{s_x} + \vec{s_y} + \vec{s_z}) \\ &= \vec{F_x} \bullet \vec{s_x} + \vec{F_y} \bullet \vec{s_y} + \vec{F_z} \bullet \vec{s_z} = W_x + W_y + W_z \end{aligned} \tag{S7}$$

The total work W is the inner product of the force $\vec{F}$ and the displacement $\vec{s}$. From another perspective, the total work W can be decomposed into the work done along the three coordinate axes. This has an analogy with the 3D-RoPE attention.

If we interpret the 3D-RoPE attention as the interaction between residues, then based on the long-term decay property of the 1D-RoPE[11], the 3D-RoPE, which can be seen as the sum of three 1D-RoPE, will also exhibit long-range decay. In this way, the 3D attention between residues will weaken as the distance increases, which nicely captures the mechanism of the inter-residue interactions.

Finally, 3D-RoPE is applied together with local attention, which ensures good extrapolation capabilities. Specifically, the number of tokens processed in each attention operation during both the training and testing phases remains consistent (i.e., limited to the nearest *k* residues in space). Even when the spatial dimensions of a tested map become very large, the local attention constrains each residue to attend to its neighboring residues only. As a result, the relative positional distances in 3D-RoPE are in a limited range, even for large maps.

S2.5 Cryo-Net training

For the convenience of narration, we restate the Eq. 3 from the main text here. The input and output of Cryo-Net in each *n*-th iteration are defined by the following equation:

$$g\left(T^{(n)}, V, S\right) = (T^{(n+1)}, \alpha^{(n+1)}, A^{(n+1)}, P^{(n+1)}, E^{(n+1)}, M^{(n+1)})$$

The frame T can be defined as (R, t), where $R \in \mathbb{R}^{r\times3\times3}$ and $t \in \mathbb{R}^{r\times3\times1}$. At the beginning of the training, $t^{(0)}$ is initialized as the Cα atomic coordinates of the training data after adding Gaussian noise $e_i \sim N\left(0, \frac{1}{\sqrt{3}}\right)$ along each dimension, and the rotation matrix $R^{(0)}$ is randomly initialized [1].

During the training process, a Cα atom is randomly selected from each PDB [12] data sample, and the 200 nearest residues in the spatial region are cropped. Inspired by ModelAngelo, 10% of the residues are randomly replaced with peptide chains of length 2-5. The purpose of this is to simulate the potential redundancy in the Cα atom coordinates output by U-Net, i.e., there may be no corresponding residue with deposited structure around the output nodes. In this case, the output *M* of the network is used to determine whether the nodes output by U-Net are redundant or not.

S2.6 CryoAtom_AF3

CryoAtom_AF3 is an integrated protocol that combines multiple tools (CryoAtom, UniDoc [13], AF3 and US-align) to leverage their complementary strengths. First, we use the following command to generate an initial model using CryoAtom.

- cryoatom build -v MAP.mrc -s SEQ.fasta -o init_model

Then, we predict the protein structure using the AF3 web server based on the sequence. We then use the UniDoc [13] command to split the AF3 predicted model into individual domains, creating a template library for the subsequent steps.

- python Run_UniDoc_from_scratch_structure.py -i af3_model.pdb -c chain -o output

Next, we use the initial model generated by CryoAtom to guide the reassembly of the protein structural units from the AF3 predicted model. Specifically, we use a greedy algorithm that maximizes the TM-score, and leverage US-align to sequentially align the structural units from the template library to the CryoAtom initial model. The command to run this process is as follows:

- cryoatom assemble --td template_dir --c init_model_raw.cif

Finally, we run the following command to extract the Cα atoms of the reassembled model as input for the Cryo-Net refinement.

- cryoatom build -r cryoatom_assemble.cif -v MAP.mrc -s SEQ.fasta -o final_model

We can also run the above process again using the original (not domain-split) AF3 predicted model as the template. This means we now have a total of three models (including the initial CryoAtom model). We will then select the best model based on the CryoAtom confidence scores, choosing the model with the most residues having a confidence score > 50.

S3. Supplementary figures

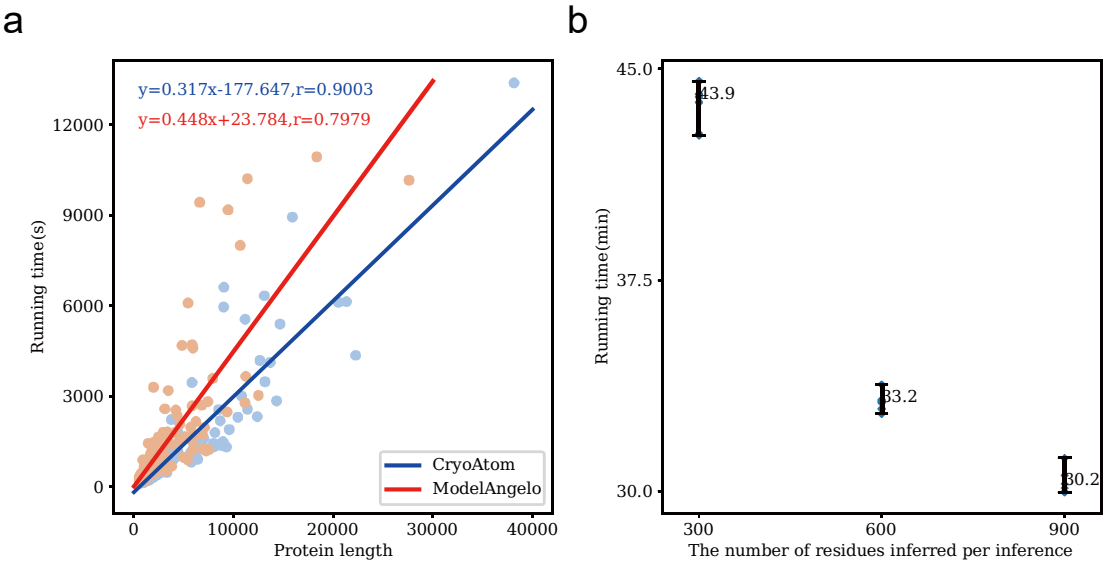

Fig. S1. Computational efficiency of CryoAtom. (a) Running time for building 177 maps using CryoAtom and ModelAngelo. (b) Running time for an example protein (EMD-29327, PDB ID: 8FNV) under different memory configurations (15GB, 19GB, 24GB). To eliminate the effects of randomness, each configuration was run six times. Error bars indicate median ±1.0 standard deviations.

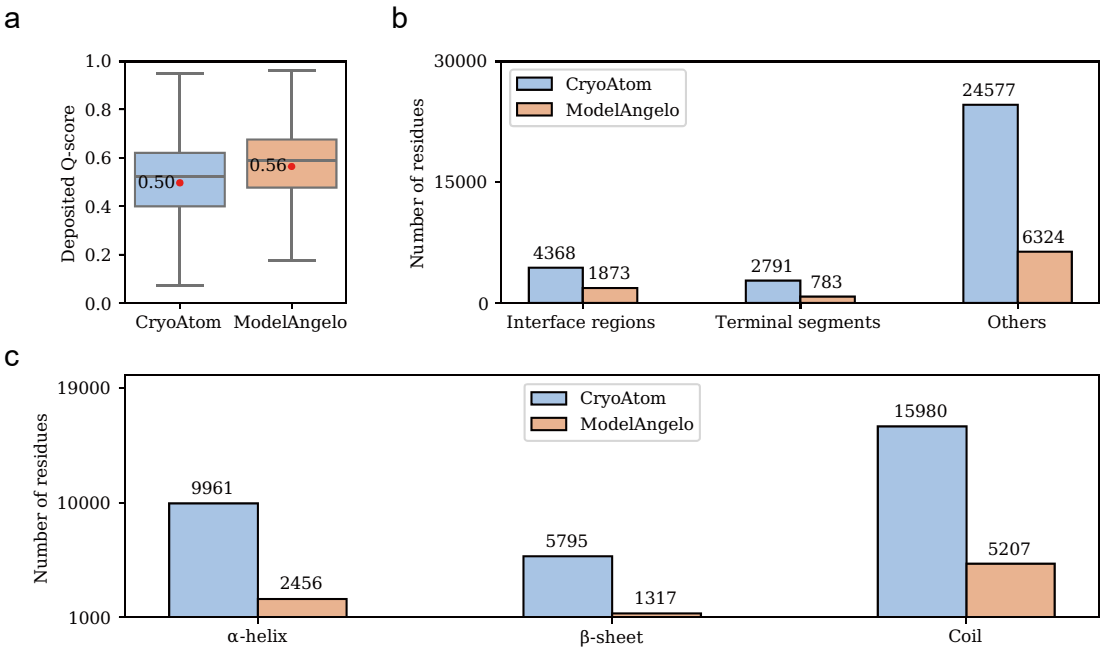

Fig. S2. Analyses of the unique TPs modeled by CryoAtom (31,736 TPs) and ModelAngelo (8,980 TPs) on 177 high-resolution maps. The unique TPs of CryoAtom and ModelAngelo are mentioned in Fig. 2f. (a) Box plots of the deposited Q-score (CryoAtom, n=31736; ModelAngelo, n =8980), in which the red dots represent the mean values. The center, lower and upper lines in each box indicate the median, the first quartile and the third quartile, respectively. The whiskers extend to the most extreme data points that are within 1.5 times the interquartile range (IQR) from the first and third quartiles. Data points beyond this range are considered outliers. (b) Bar plots of the number of residues in the interface regions, terminal segments and others, for the unique TPs in the CryoAtom and ModelAngelo models. (c) Bar plots of the unique TPs of CryoAtom and ModelAngelo belonging to different secondary structure categories.

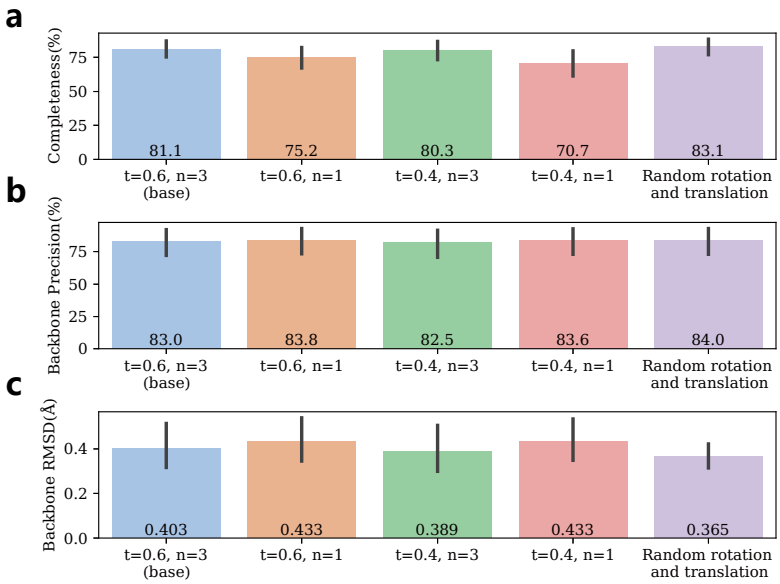

Fig. S3. Randomly selected 33 density maps from the sequence non-redundant test set to test CryoAtom's sensitivity to hyperparameters and rotation-translation. The parameter *t* represents the threshold for predictions made by the classification network in Stage 1. The parameter *n* represents the number of recycling rounds in Stage 2. Random rotation and translation refer to the density maps. CryoAtom will use the transformed density map as input and compare the results with the native model. (a-c) Bar plots comparing the average completeness, backbone precision, backbone RMSD (n=33). The height of each bar graph represents the average value, while the error bars are calculated based on the 95% confidence interval.

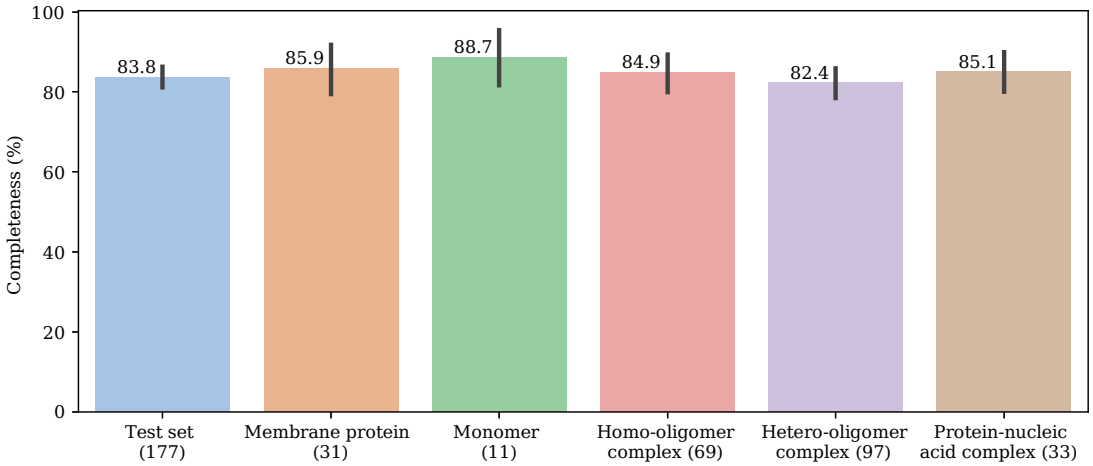

Fig. S4. The performance of CryoAtom on 177 high-resolution maps testing set for different biological systems proteins. Bar plots comparing the average completeness. The height of each bar graph represents the average value, while the error bars are calculated based on the 95% confidence interval.

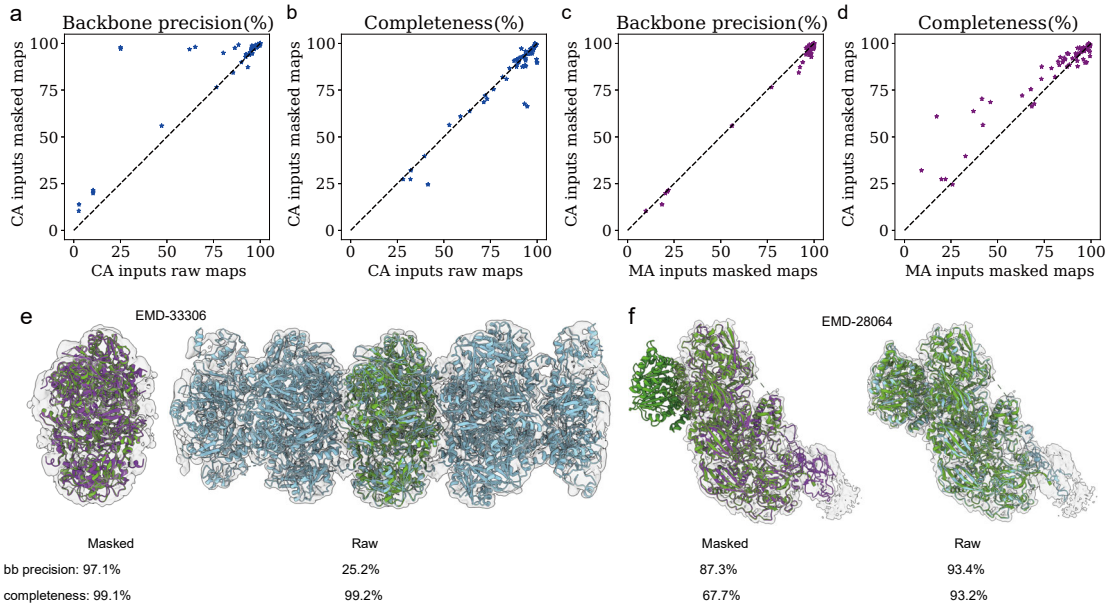

Fig. S5. The impact of map masking. (a) Head-to-head comparison of the model backbone precision. (b) Head-to-head comparison of the model completeness. (c)/(d) Head-to-head comparison of the model backbone precision/completeness between CryoAtom and ModelAngelo by inputting masked maps. (e) Appropriate masking can screen out regions unrelated to the deposited structure protein. The gray surface is the density map. The green cartoon represents the deposited structure. The purple/blue cartoon is the CryoAtom model constructed using the masked/raw map. (f) Inappropriate masking can lead to incorrect atomic model. The coloring scheme is the same as in (e).

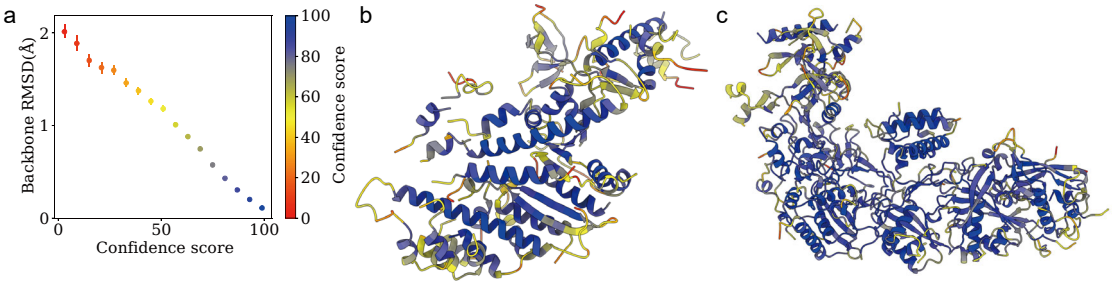

Fig. S6. Correlation analysis between CryoAtom confidence score and backbone RMSD. (a) Confidence score bin size is 6, and the error bars represent the 99% confidence interval of the mean on a per-residue basis. (b-c) CryoAtom models colored by confidence score for the maps EMD-28080 (b, PDB ID: 8EFD, reported resolution 3.8 Å) and EMD-33678 (c, PDB ID: 7Y82, reported resolution 2.83 Å).

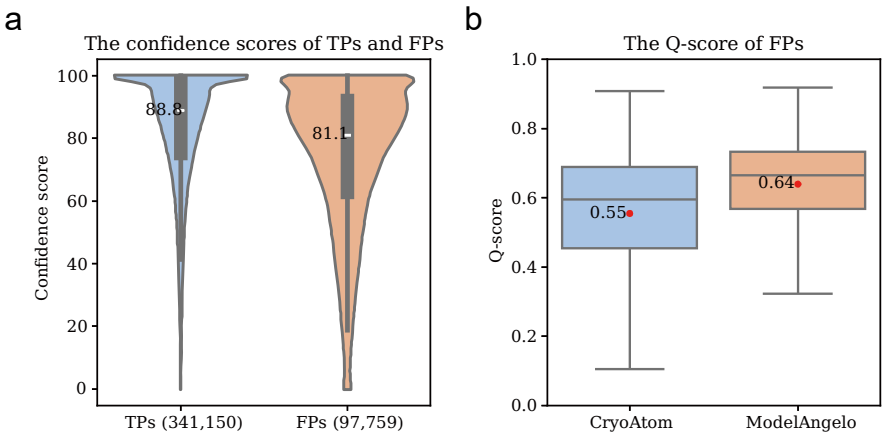

Fig. S7. Quantitative analyses of the possibility of misjudgment for certain FPs. (a) The distribution of confidence scores of the CryoAtom models for the 177 maps. TPs: true positives, FPs: false positives. Note that the FPs are possible to be misjudgments, as illustrated by the example in Fig. 3. (b) Box plots of the Q-score of FPs (CryoAtom, n=97759; ModelAngelo, n =64797), in which the red dots represent the mean values. The shape of the violin plot (a) indicates the distribution. The center, lower and upper lines in each box (a-b) indicate the median, the first quartile and the third quartile, respectively. The whiskers extend to the most extreme data points that are within 1.5 times the interquartile range (IQR) from the first and third quartiles. Data points beyond this range are considered outliers.

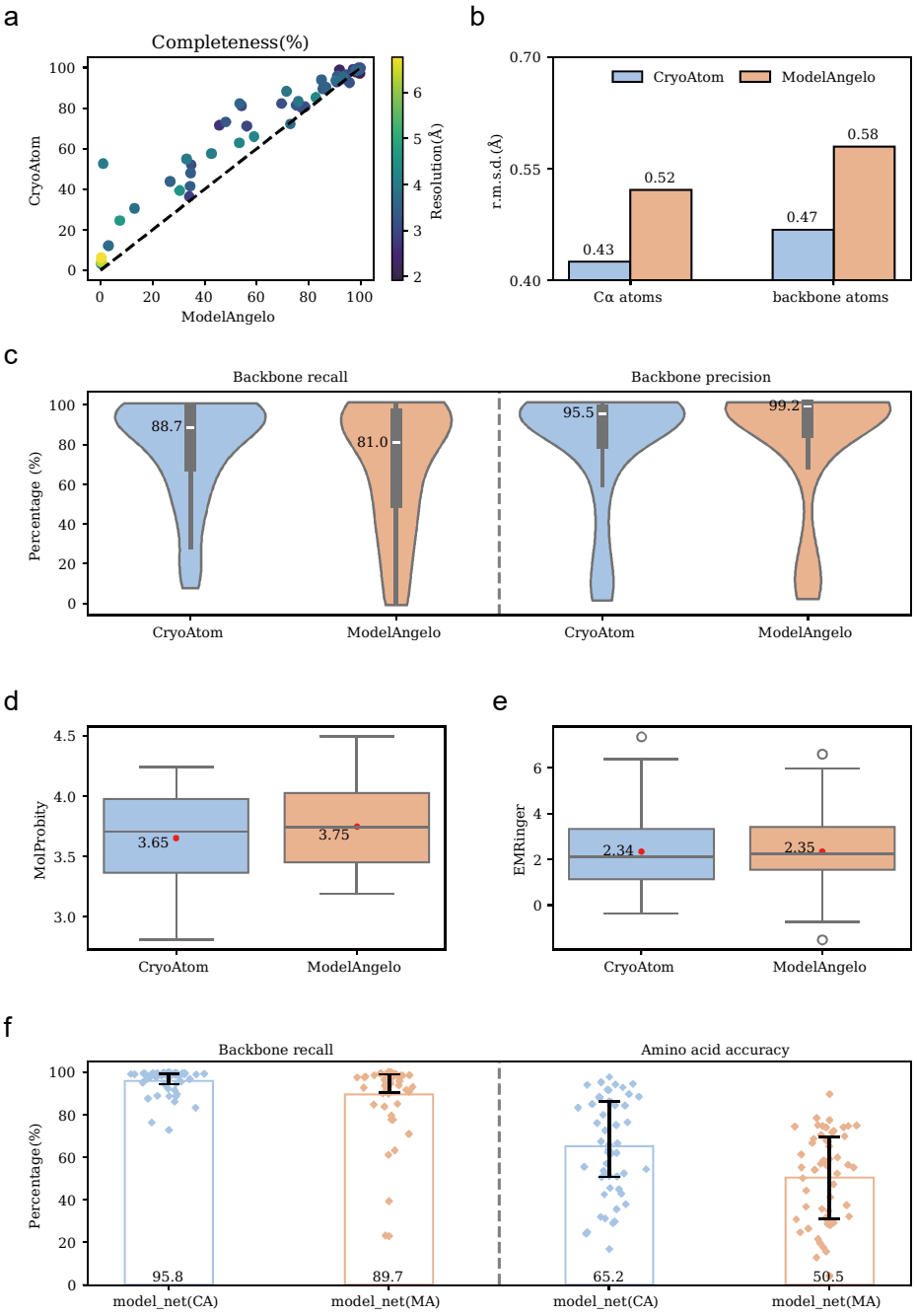

Fig. S8. Comparison of CryoAtom and ModelAngelo on 54 non-redundant density maps. (a) Head-to-head comparison of the model completeness. (b) Bar plots comparing the average RMSDs of Cα atoms and backbone atoms. (c) Violin plots of the backbone recall/precision (n=54). (d) Box plots of the MolProbity score, in which the red dots represent the mean values (n=54). (e) Box plots of EMRinger, in which the red dots represent the mean values (n=54). The shape of the violin plot (c) indicates the distribution. The center, lower and upper lines in each box (c-e) indicate the median, the first quartile and the third quartile, respectively. The whiskers extend to the most extreme data points that are within 1.5 times the interquartile range (IQR) from the first and third quartiles. Data points beyond this range are considered outliers. (f) Backbone

recall and amino acid accuracy of the intermediate models (n=54). Error bars indicate ±1.0 standard deviations.

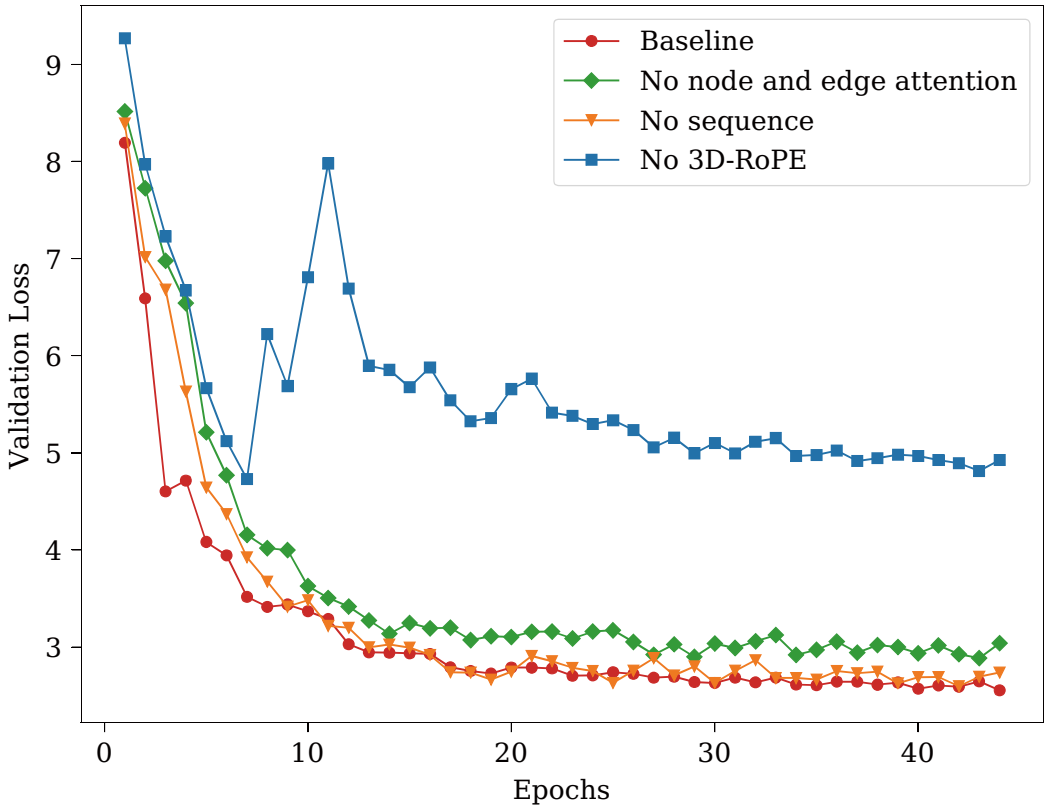

Fig. S9. The curves of loss function under four different configurations.

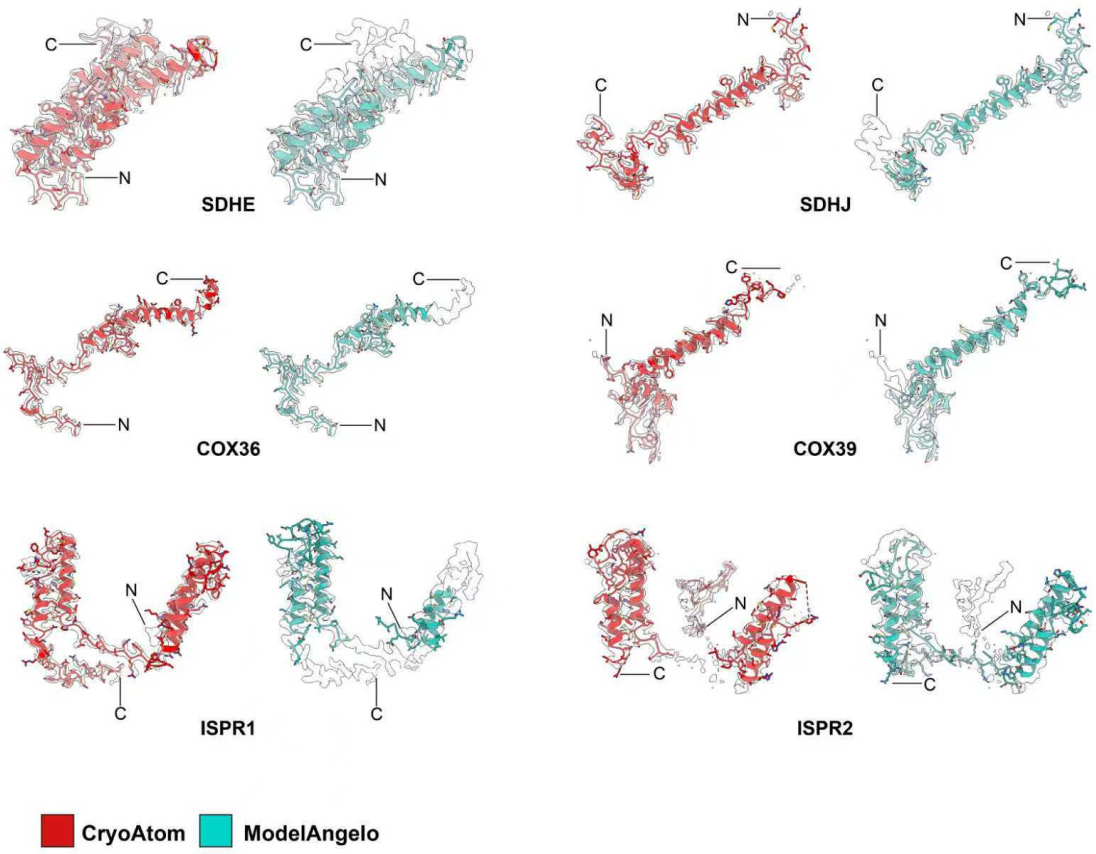

Fig. S10. Modeling evaluation by the CryoAtom and ModelAngelo in low-resolution regions. Six protein models of the mitochondrial respirasome II2-III$_2$-IV2 [14] are compared and shown with their corresponding densities. Termini are indicated for each protein.

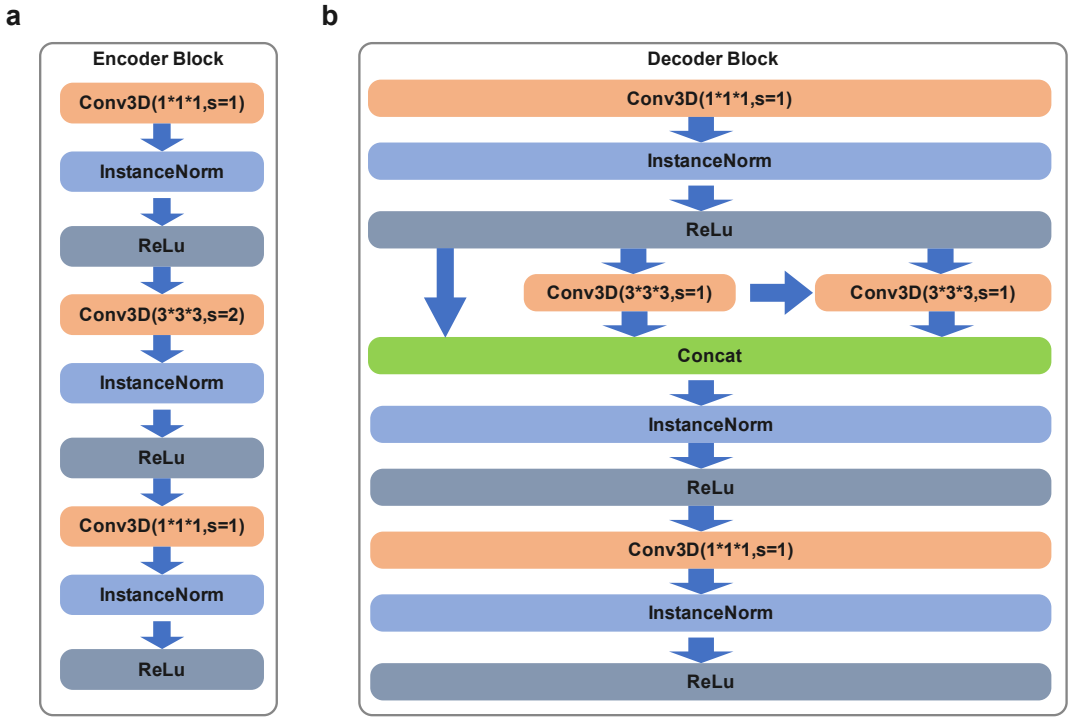

Fig. S11. Architecture of U-Net. (a) Encoder module and the downsampling module are integrated together, and a bottleneck architecture[6] is adopted. (b) Decoder module adopts the Res2Net architecture [7, 8].

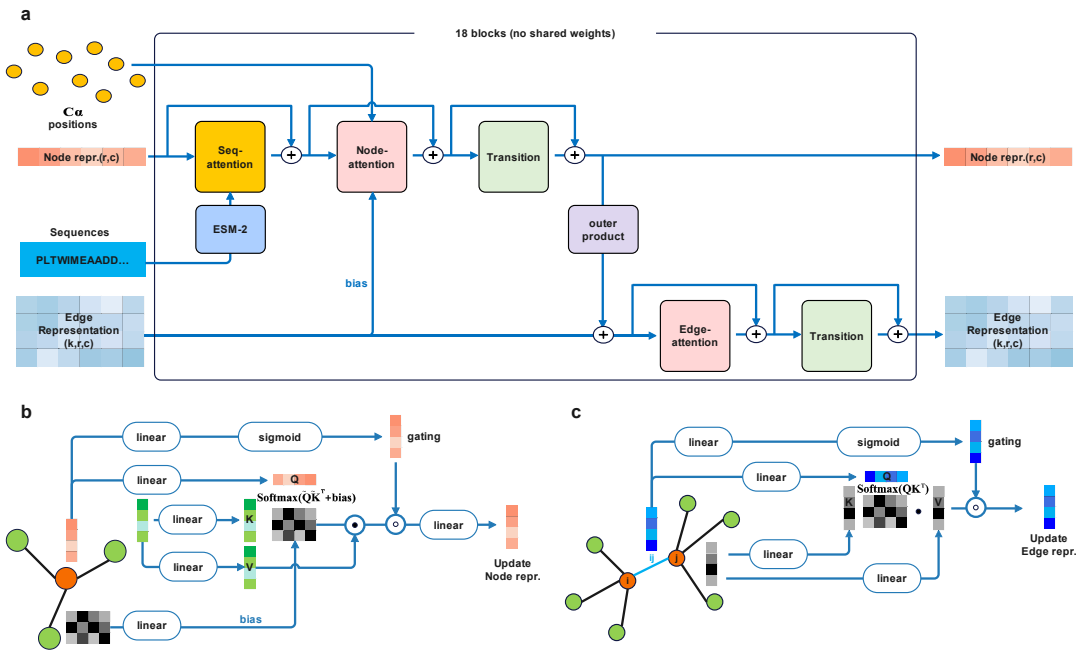

Fig. S12. Architectural details of the encoder network. (a) Cryo-Former module. (b) node attention layer. (c) edge attention layer.

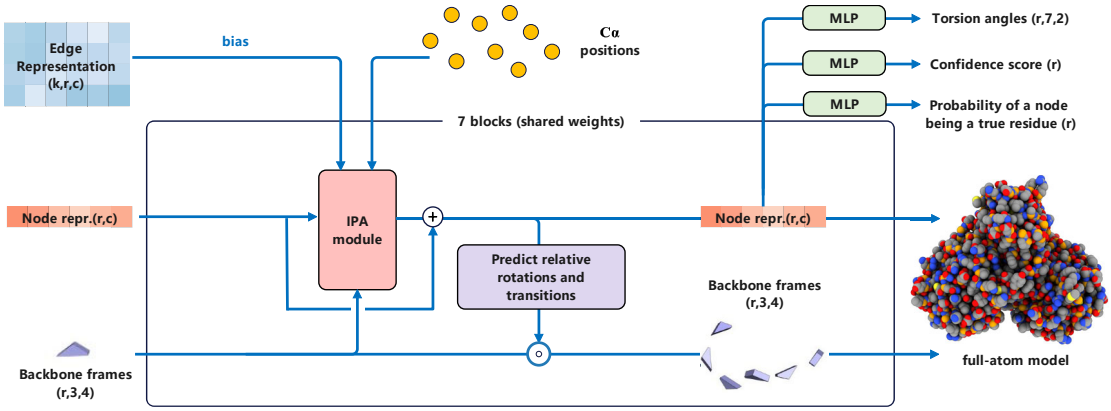

Fig. S13. Architectural details of the decoder network Structure Module. This module is similar to the structure module in AF2 [10]. The difference is that the input here includes additional node position information (i.e., the Cα positions). The updated node representations are passed through separate MLPs to output *T, α, P, M* in Eq. 3.

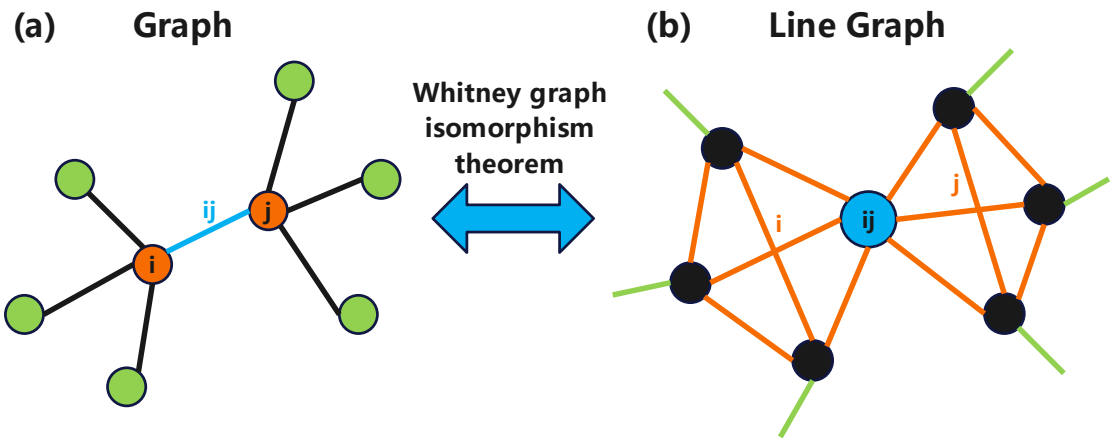

Fig. S14. Relationship between node attention and edge attention. A graph (a) can be converted into a line graph (b), and vice versa. According to the Whitney graph isomorphism theorem, except for a very small number of special cases, the line graph uniquely determines the original graph. These two types of graphs have a certain duality. The edge attention to the edge *ij* of the graph on the left is equivalent to the node-attention to the *ij* node of the line graph on the right.

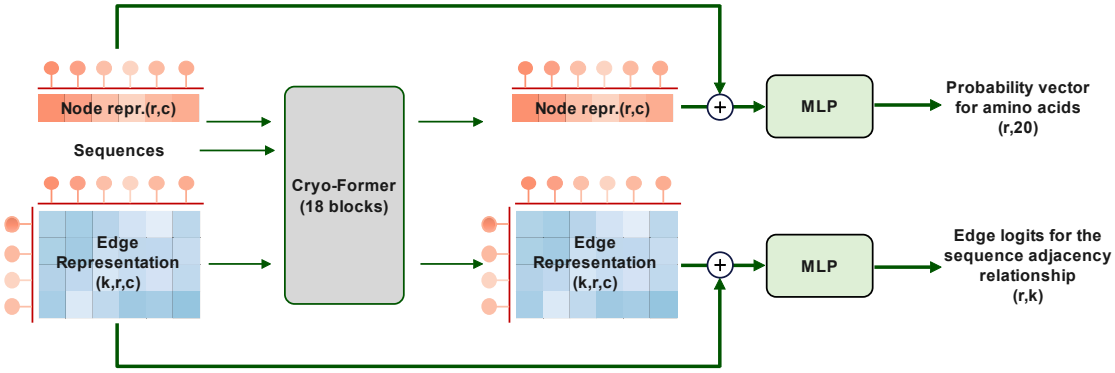

Fig. S15. Detailed information for the Cryo-Former Module. The original node representation and the updated node representation are concatenated and passed through a separate MLP to predict the probability vector over the 20 types of amino acids for each node (i.e., A in Eq. 3). The original edge representation and the updated edge representation are concatenated and passed through another separate MLP to predict the edge connectivity (i.e., E in Eq. 3).

Supplementary References

1. Jamali, K. *et al.* Automated model building and protein identification in cryo-EM maps. *Nature* **628**, 450-457 (2024).
2. Kuhn, H.W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**, 83-97 (1955).
3. Zhang, C., Shine, M., Pyle, A.M. & Zhang, Y. US-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature Methods* **19**, 1109-1115 (2022).
4. Turner, J. *et al.* EMDB—the Electron Microscopy Data Bank. *Nucleic Acids Research* **52**, D456-D465 (2024).
5. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation, 234-241 (Springer International Publishing, 2015).
6. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.C. in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 4510-4520 (2018).
7. Gao, S.H. *et al.* Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 652-662 (2021).
8. Su, H. *et al.* Improved Protein Structure Prediction Using a New Multi-Scale Network and Homologous Templates. *Adv Sci (Weinh)* **8**, e2102592 (2021).
9. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. **379**, 1123-1130 (2023).
10. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583-589 (2021).
11. Su, J. *et al.* RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing* **568**, 127063 (2024).
12. Burley, S.K. *et al.* RCSB Protein Data Bank (RCSB.org): delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic Acids Research* **51**, D488-D508 (2023).
13. Zhu, K., Su, H., Peng, Z. & Yang, J. A unified approach to protein domain parsing with inter-residue distance matrix. *Bioinformatics* **39**, btad070 (2023).
14. Wú, F. *et al.* Structure of the II2-III2-IV2 mitochondrial supercomplex from the parasite Perkinsus marinus. *bioRxiv*, 2024.2005.2025.595893 (2024).