MODELANGELO: AUTOMATED MODEL BUILDING IN CRYO-EM MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Electron cryo-microscopy (cryo-EM) produces three-dimensional (3D) maps of the electrostatic potential of biological macromolecules, including proteins. At sufficient resolution, the cryo-EM maps, along with some knowledge about the imaged molecules, allow *de novo* atomic modelling. Typically, this is done through a laborious manual process. Recent advances in machine learning applications to protein structure prediction show potential for automating this process. Taking inspiration from these techniques, we have built *ModelAngelo* for automated model building of proteins in cryo-EM maps. ModelAngelo first uses a residual convolutional neural network (CNN) to initialize a graph representation with nodes assigned to individual amino acids of the proteins in the map and edges representing the protein chain. The graph is then refined with a graph neural network (GNN) that combines the cryo-EM data, the amino acid sequence data and prior knowledge about protein geometries. The GNN refines the geometry of the protein chain and classifies the amino acids for each of its nodes. The final graph is post-processed with a hidden Markov model (HMM) search to map each protein chain to entries in a user provided sequence file. Application to 28 test cases shows that ModelAngelo outperforms the state-of-the-art and approximates manual building for cryo-EM maps with resolutions better than 3.5 Å.

1 INTRODUCTION

Following rapid developments in microscopy hardware and image processing software, for favourable samples, cryo-EM structure determination of biological macromolecules is now possible to atomic resolution (Nakane et al., 2020; Yip et al., 2020). For many other samples, such as large multi-component complexes and membrane proteins, resolutions around 3 Å are typical (Cheng, 2018). In this method, transmission electron microscopy images are taken of many copies of the same molecules, which are frozen in a thin layer of vitreous ice. Each field of view contains hundreds of two-dimensional projections of the electrostatic potential of such molecules in unknown orientations. Radiation damage limits the amounts of electrons that can be used for imaging, which results in extremely low signal-to-noise ratios in the images. Consequently, averaging over many copies is necessary, and one often acquires thousands of field of views for a given sample. Dedicated softwares, like RELION (Scheres, 2012) or cryoSPARC (Punjani et al., 2017), implement iterative optimization algorithms to retrieve the unknown orientation of each molecule, and perform 3D reconstruction to obtain a voxel-based map of the underlying molecular structure.

Provided the cryo-EM map is of sufficient resolution, it is then interpreted in terms of an atomic model of the corresponding macromolecular structures. Many samples contain only proteins; other samples also contain other biological molecules, like lipids or nucleic acids. Proteins are linear chains of amino acids, or residues. There are twenty different amino acids that serve as the primarily building blocks of proteins. All of these amino acids have four heavy (non-hydrogen) atoms that make up the protein main chain. The different amino acids have different numbers, types and geometrical arrangements of their side-chain atoms. The smallest amino acid, glycine, has no heavy side chain atoms; the largest amino acid, tryptophan, has ten heavy side chain atoms. Typical proteins range in size from tens to more than a thousand residues. The genes of an organism determine the order of the amino acids in each protein. Often, but not always, the electron microscopist knows which proteins are present in the sample. The task at hand is to build the atomic model, which identifies the positions of all atoms for all proteins that are present in the reconstructed map. For each

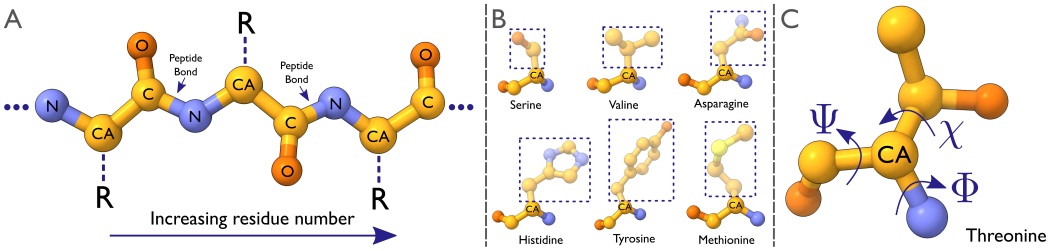

Figure 1: **(A)** shows a backbone where the chain is ordered from left to right. Arrows mark the peptide bonds between the C atom of one residue and the N atom of the next. **(B)** shows six amino-acids, pointing out the differences in the side chains, marked by the outline. **(C)** shows the Φ and Ψ angles of the backbone as well as the additional rotatable bond of the side chain for the amino-acid threonine.

residue, there are two rotational degrees of freedom in the conformation of its main chain. Distinct orientations of the side chains provide additional conformational possibilities, the number of which depends on the type of amino acid (figure 1).

Atomic model building in cryo-EM maps is typically done manually using 3D visualisation software, (e.g. Emsley et al., 2010; Pettersen et al., 2021), followed by refinement procedures that optimize the fit of the models in the map, (e.g. Murshudov et al., 2011; Croll, 2018; Liebschner et al., 2019). Building a reliable atomic model *de novo* in the reconstructed cryo-EM map is considered to be difficult for maps with resolutions worse than 4 Å. Although the task is more straightforward for maps with resolutions better than 3 Å, it still typically requires large amounts of time and a high level of expertise.

Machine learning has recently achieved a major step forward in protein structure prediction (Jumper et al., 2021; Baek et al., 2021). In these approaches, the sequence information of proteins and their evolutionary related homologues is used to predict the atomic structure of proteins without the use of experimental data. In addition, protein language models, which are trained in an unsupervised fashion on the amino acid sequences of many proteins, have also provided useful results in protein structure prediction (Lin et al., 2022; Wu et al., 2022a). These successes raise the question whether improvements to automated model building could be achieved by taking inspiration from novel machine learning approaches.

In this paper, we present a machine learning approach to automated model building in cryo-EM maps. Our approach combines modern GNN architectures and protein language models with new techniques of incorporating the voxel-based information from cryo-EM maps alongside the amino-acid sequence and structural information for automated model building. In order to accomplish this, we implemented a pipeline that first seeds the map with the approximate locations of the individual residues using a CNN, and then uses a GNN to refine the full atom positions of each residue. This GNN has novel components that allow it to use information from the different modalities of text (the amino-acid sequence), 3D volume (the cryo-EM map), as well as graphs (the positions of the atoms) in one integrated neural network. Finally, a series of post-processing steps seek to digest the model and present it in a human-readable format to users.

2 Prior Work

Automated model building. Automated approaches for atomic modelling in the related experimental technique of X-ray crystallography have existed for many years (for example, Perrakis et al., 1999; Cowtan, 2006; Terwilliger et al., 2008). Although some of these approaches have also been applied to the more widely applicable technique of cryo-EM structure determination, their overall impact on the field has, thus far, been modest.

More recently, Deeptracer (Pfab et al., 2021), the first deep learning approach for automated atomic modelling in cryo-EM maps, was reported to outperform previously existing approaches. Deeptracer uses a U-Net (Ronneberger et al., 2015), coupled with some heuristics, to construct an atomic model *de novo* in the cryo-EM map. In contrast to our work, Deeptracer does not integrate the sequence

information with the U-Net, and it does not use a graph representation of the protein chain during model refinement. Instead, Deeptracer treats the entire problem as a segmentation and classification problem. Thereby, it also does not have support for refining already built models or performing multiple recycling steps. Although Deeptracer predicts amino acid types for each residue, it only builds atoms for the main chains.

There have also been reports to use protein structure prediction programs, like AlphaFold2 (Jumper et al., 2021), to morph their output predictions to fit the cryo-EM map (He et al., 2022; Terwilliger et al., 2022). Such approaches are likely to propagate errors in the structure prediction. Thus, it seems sensible to design a neural network approach that integrates both the cryo-EM map and the sequence and protein structure primitives to produce a more reliable structure. This is the approach of ModelAngelo.

GNNs for proteins. A number of different approaches to modelling proteins with GNNs have been proposed recently. This includes modelling the protein with torsion angles (Jing et al., 2022), SE(3) equivariant graph neural networks (Ganea et al., 2022), and SE(3) invariant GNNs (Dauparas et al., 2022). In our approach, the ordering and connectivity of residues are unknown and have to be inferred, hence representations that require this to be known *a priori*, such as the torsion angle representation, are inappropriate. Furthermore, the relative orientation of the model and the cryo-EM map is important in model building, which makes SE(3) invariant representations ill-suited. Thus, the most natural graph representation is an SE(3) equivariant one. In this project, we choose the backbone frame representation that was first introduced in AlphaFold2, but is now also used in other protein prediction networks (see Wu et al., 2022a; Lin et al., 2022).

3 METHODS

3.1 RESIDUE SEGMENTATION

3.1.1 TRAINING DATASET

The first step in ModelAngelo is to identify where individual residues are placed. For proteins in our approach, this is modelled as the position of the C^α atom [1] (see figure 1A) of each amino acid. This part of the pipeline is formulated as a straight-forward segmentation problem. That is, the cryo-EM map $V \in \mathbb{R}^{N^3}$, where the N is the number of voxels, has an associated binary target T, where 1 represents the existence of a C^α atom in the voxel and 0 the lack of it. Since the minimum distance of two C^α atoms is 3.8 Å, resampling the cryo-EM voxel maps with a pixel size of 1.5 Å ensures that there is no voxel that contains more than one such atom. The goal then becomes to train a neural network $f_\theta(V) \approx T$. We did this in a supervised manner using online deposited pairs of cryo-EM volumes and PDB models.

The dataset starts with 6351 cryo-EM maps with resolution higher than 4 Å, downloaded from the EMDB (Lawson et al., 2016) on 01/04/2022, and the corresponding PDB files that were downloaded from RCSB (Burley et al., 2021). A portion of these were manually checked for orientation issues, large unmodelled regions, and existence of large modelled regions that do not correspond to the cryo-EM map. This led to $\sim$700 manually curated pairs for the first round of training. Then, using the first trained model, we were able to automatically detect issues with the rest of the map-structure pairs and prune them to $\sim$3200 structures. The pruning was based on a cutoff of 70% precision and 70% recall of the model output C^α positions compared to the ground-truth PDB coordinates. This produced the dataset that was used for training the model.

3.1.2 TRAINING

For insight into the difficulty of this machine learning task compared to regular segmentation problems, we examine PDB entry 7PT7. This is a protein with 8455 modelled amino-acid residues. The cryo-EM map, after being rescaled to 1.5 Å pixel size, has 16,777,216 voxels. That means that only 0.05 % of the voxels have a C^α atom. These are typical numbers, illustrating that this task is a highly imbalanced classification problem. Therefore, conventional practices for segmentation are

[1] Atom names in this document follow the PDB (Protein Data Bank) naming convention. See the PDB atomic coordinate and bibliographic entry format description

likely not to give good results. Instead, we up-weighted the loss of voxels with C^α atoms and used a combination of loss functions to get an acceptable trade-off between precision and recall. In order to balance the loss from the empty voxels and the C^α containing ones, we used the following formula:

$$w_x = \chi(x)\frac{N^3 - C}{C} + (1 - \chi(x)) \tag{1}$$

where w_x is the weight for voxel x, χ is the characteristic function of whether x contains a C^α atom, N^3 is the number of voxels, and C is the number of C^α atoms in the map. This formula ensures that half of the loss of a map comes from empty voxels and half from the C^α atom containing voxels. Another technique that we used to ameliorate the issue with class imbalance is to use focal loss (Lin et al., 2017b) instead of the usual binary cross entropy loss. We found that also using Tversky loss (Hashemi et al., 2018) in the later parts of training allowed higher recall at the expense of precision. Lastly, to improve generalization, we applied data augmentation schemes to the cryo-EM maps. We added random, spectrum-scaled white noise to every map, and performed random sharpening/dampening by sampling a B-factor (see Rosenthal & Henderson, 2003) from a uniform distribution between -30 $Å^2$ and 30 $Å^2$. Lastly, we applied random rotations that are integer multiples of 90° to both the cryo-EM map and the targets in order to have the model learn different orientations, while avoiding interpolation effects.

3.1.3 Network Architecture

The segmentation model is a residual network (He et al., 2016) inspired by the Feature Pyramid Network (Lin et al., 2017a), which we modified to improve its performance for this problem. We changed all convolutions to 3D convolutions and we changed batch normalization layers (Ioffe & Szegedy, 2015) to instance normalization layers (Ulyanov et al., 2016), due to improved results. We did not see a significant deviation in results between ReLU and LeakyReLU (Xu et al., 2015), so we switched to ReLU for its improved computational efficiency. We also shifted the network parameters from the low resolution part of the model to the high resolution part, and we changed the order of operations, so that global information about structure is more directly integrated into the model at an early stage. In this manner, large scale structural features, which are recognized at lower resolutions, become easier to integrate with the classification task.

3.2 All-atom modelling

3.2.1 Network Architecture

Starting from approximate positions for the C^α atoms, we need to obtain a complete atomic model. This consists of a few separate tasks. Since the input to the network is an unordered collection of C^α atoms, the amino-acid types, the chain directions, the side chain placements, and the correspondence to the given sequence need to be determined. In order to accomplish this, all residues are represented with a backbone affine frame and a series of torsion angles, similar to AlphaFold2 (Jumper et al., 2021), with orientations initialized randomly and then optimized using the output of a graph neural network (GNN), as can be seen in figure 2.

Unlike AlphaFold2, which already has access to the residue-to-sequence relationship due to the nature of the protein structure prediction problem, our GNN cannot make any assumptions about the order of its input nodes. Therefore it also needs an amino-acid classification probability for each residue, along with learning the correspondence of the graph nodes to the sequence. The GNN consists of three main modules, all based on the attention algorithm (Bahdanau et al., 2014), that are stacked 8 times in order to sequentially optimize the output structure.

The first of these modules is the Cryo-EM Attention module, which allows the GNN to look at the density around each C^α atom with convolutional neural networks (CNN), as well as the density of neighbouring nodes, to update its representation. This is accomplished with a novel mix of a graph-based attention module and feature extraction using CNNs. A cube centered at each node is interpolated from the cryo-EM density and the orientation of this cube is defined by the backbone affine frame of each node. Similarly, rectangles of cryo-EM density are interpolated along the vectors that connect each node to its closest k neighbours. The query and value vectors are generated by the feature vector of each node, while the key vectors are generated from the cryo-EM rectangles between neighbours. Finally, the features generated from the centered cubes are concatenated with

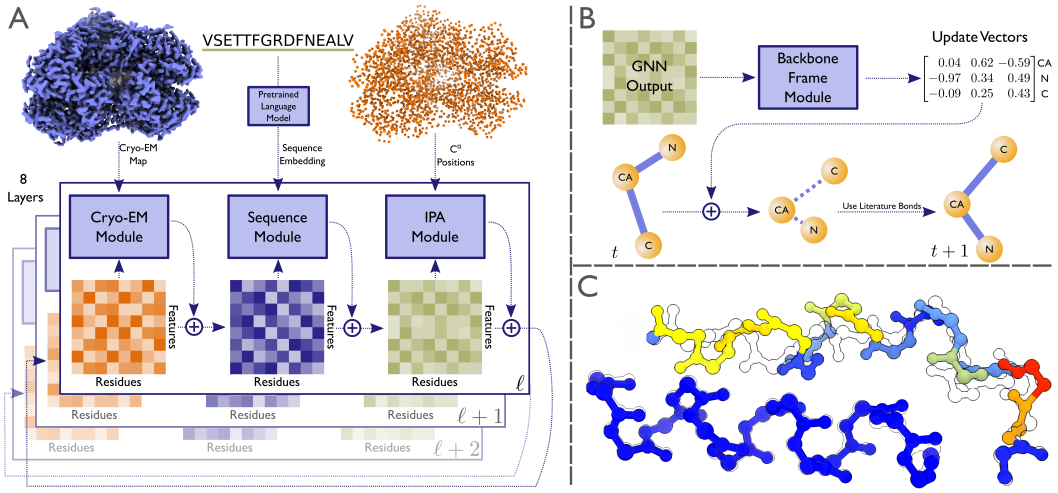

Figure 2: **(A)** shows the schematic of the graph neural network of ModelAngelo and how the 8 layers of the GNN iteratively refine the feature vectors per residue that are used to predict the atomic model. **(B)** illustrates how the Backbone Frame module updates the positions of the backbone atoms by first predicting a set of shift vectors, updating the positions, and then re-projecting the atoms back to literature bond lengths and angles. Finally, **(C)** contains two examples of high confidence (dark blue colour) and low confidence (yellow and red) predicted backbone regions for PDB entry 7Z1M. The confidence measure is a good predictor of fit to the deposited backbone model (shown in outline).

the attention output and projected to create the new features. This allows the network to see how strong the density between two nodes is before it makes decisions to mix features (see algorithm 1 in A.4).

The second module is the Sequence Attention module, which lets each node search against the input sequence embedding to find the relevant part that matches its features. This is a conventional encoder-only transformer module, similar to that used in Devlin et al. (2018). The sequence is first embedded using a pre-trained protein language model (Rives et al., 2021; Lin et al., 2022). We only use the primary sequence, instead of multiple sequence alignments (MSAs). Generally, MSAs improve results of protein prediction algorithms by giving them access to co-evolutionary information (Jumper et al., 2021). In fact, to match the results of networks with MSAs, algorithms that use the primary sequence need to have many more learnable parameters (Rao et al., 2021). However, we believe ModelAngelo does not need them for a number of reasons. Firstly, there is evidence to show that MSAs are capable at elucidating the global arrangement of the protein fold, but for local refinements they are less useful (Roney & Ovchinnikov, 2022). Since we already know the global fold from the cryo-EM map alone, MSAs are not going to offer much more information. Secondly, generating and processing MSAs creates computational costs that have prompted new protein prediction algorithms to focus on single sequence prediction (Lin et al., 2022; Wu et al., 2022b).

Finally, there is the Spatial Invariant Point Attention (IPA) module which allows the network to update its representation based on the geometry of the nodes in the graph. This is inspired by the similarly named module in AlphaFold2, however simplifications have been made to better fit the problem at hand. Each node predicts query points based on its current representation in its own local affine frame, these points are transformed into the global affine frame, the distance is calculated between each node's query points and its neighbours, and based on the sum of the distances of the query points to the nodes, each node gets an attention score that is used to update its representation. Essentially, it queries parts of the graph where it expects specific nodes to be and then uses the distance of the neighbouring nodes to its query point to collect information from the other nodes (see algorithm 2 in A.4).

Initial features of the GNN only include the distance of each node with its 20 nearest neighbours, which are encoded using positional embeddings as in Vaswani et al. (2017). The features are then fed into each of the modules and optimized with a residual connection and the output of each module.

Since these modules are applied sequentially multiple times, the representations from each module allow other modules to gradually extract more information from their inputs. For example, using the cryo-EM density, the network is able to find a better orientation for its backbone as well as a more accurate set of probabilities for its amino acid identity, which lets it search the sequence more accurately with the sequence attention module. This process of improvement continues while the positions of the atoms also get optimized using these representations through the application of the Backbone Frame module.

The Backbone Frame module (seen in figure 2B) takes as input the representation of each graph node that is the result of the sequential operation described above, and outputs three vectors that describe the change in position of the C^α, C, and N atom positions with respect to the network's current backbone affine frame. The shift in position is applied to the backbone atoms and the new backbone affine frame is calculated using Gram-Schmidt, similar to Algorithm 21 in AlphaFold2 (Jumper et al., 2021). After the backbone affine frame has been defined, we then use the known bond lengths and angles to get new positions for the C and N atoms.

3.2.2 TRAINING

Since this model does a series of tasks, there is no single loss function. Instead, the loss is split across different tasks and is optimized jointly with gradient descent. Most losses are calculated at each intermediate layer of the GNN so that it is able to learn the correct structure as early in the layers as possible. The most important losses are as follows: C^α root mean squared deviation (RMSD) loss, backbone RMSD loss, amino-acid classification loss, local confidence score loss, torsion angles loss, and full atom loss. A full definition of all losses are available in Appendix A.1.

The main training loop consists of taking a PDB structure, extracting just the C^α atoms, distorting them with noise, initializing the backbone frames for each node randomly, and then having the network predict the original PDB structure. Cryo-EM maps are augmented similar to segmentation training, described in section 3.1.2. Because the initial C^α positions are noisy, with an RMSD to the deposited model of 0.9 Å on average, one important loss function is

$$\mathcal{L}_{C^\alpha} = \frac{1}{N} \sum_i \mathrm{RMSD}(\mathbf{x}_i, f_\theta(\mathbf{x}_i + \mathbf{e}_i)) \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^3$ are the true C^α positions from the dataset, f_θ is the graph neural network, and $\mathbf{e}_i \sim \mathcal{N}(0, \frac{1}{\sqrt{3}})$. Note that $\mathbb{E}[\mathrm{RMSD}(\mathbf{e}_i, \mathbf{0})] \approx 0.9$ (this comes from the average norm of a Gaussian distributed vector). Denoising node positions has been shown to be a powerful training paradigm in other use cases as well (e.g. Godwin et al., 2021). We did not use the AlphaFold2 frame aligned point error loss as the assumed invariance to rotation in that loss is not relevant in our case, since a given cryo-EM map imparts a specific orientation for the atomic model, modulo symmetry. In addition to noise added to the C^α positions, sometimes the segmentation model misses some residues, or adds extra ones. In order to have the GNN be able to deal with these scenarios, during training 10% of residues are randomly removed and replaced with randomly generated peptides between 2-5 residues long that are placed in the model. The network is then also trained to be able to predict whether or not a node actually exists in the model, or if it is extra. All classification losses use focal loss (Lin et al., 2017b). This includes the amino-acid classification loss, the sequence match loss, and the edge prediction loss.

An important feature of this network is that it also gives a measure of its confidence in its output per residue. This is trained by having the network predict its backbone loss per residue. This output is then normalized and saved in the B-factor section of the mmCIF file it outputs. This is a helpful output because we can observe where the model is highly certain and also use it to prune non-confident regions. Generally, we observe that highly structured and high resolution parts of the cryo-EM map, especially parts with alpha helices and beta sheets, have higher confidence values, while more disordered and lower resolution areas do not (see figure 2C).

The side chain atoms are generated through prediction of their rotatable torsion angles with respect to the backbone frame (for an example, see figure 1C). We noticed better results if the network

predicted torsion angles for all 20 amino acids per residue. Then, based on the predicted amino-acid, we index into the torsion angle predictions and pick the set of angles that correspond to the residue. To train this part of the model, for each layer, the mean squared loss of the torsion angles of the target amino-acid against the true torsion angles is calculated, and at the last layer, the all-atom RMSD to the target structure is also calculated.

3.2.3 RECYCLING

The output of one round of the GNN denoises the positions of the C^α nodes and gives better orientations for the backbone frames, and we observed that a subsequent round of the GNN, starting from the output of the previous round, improved the results further. We therefore train the GNN with recycling. For every training step, we randomly pick an integer $r \in \{1, 2, 3\}$ and run the GNN $r - 1$ times with gradients turned off, and then use the output to run the GNN one more time with gradients. This allows the GNN to learn to keep the input approximately unchanged when the positions are correct. We do not recycle the GNN features so they are recalculated with the corrected positions and orientations.

3.3 POSTPROCESSING

The GNN processes the C^α atoms into a set of unordered residues. Next, we connect the residues into chains that define the full atomic model. In the strictest sense, not even the direction of the chains is defined by the GNN. However, using the fact that $\|\mathrm{C}_{t-1} - \mathrm{N}_t\|_2 < 1.4$ Å (known as peptide bonds, see figure 1A), we can combine the atomic coordinates predicted by the network as well as the edge prediction probabilities as a heuristic to connect residues. More concretely, the residues are tied so that the sum of peptide bond lengths across all nodes is minimized, ignoring links where the edge prediction is below the threshold of 0.5.

After the chains are connected, we use the amino acid prediction probabilities to construct an HMM profile. We then perform a sequence search using HMMER (Mistry et al., 2013) against the given set of sequences of the model. This is more sensitive than the approach in Deeptracer and findMySequence (Chojnowski et al., 2022), which only use the highest probability amino acid for each residue. After alignment with the sequence search, residues that correspond to a "match" state (as defined in Krogh et al., 1994) are mutated to the amino acids that exist in the sequence. Based on the sequence search, we also connect separate chains that should be connected depending on both the matched sequence gap and the proximity of the end of one chain and the beginning of the next.

Lastly, chains shorter than 4 residues are pruned and the resulting coordinates (in the format of an mmCIF file) are used as the input to the GNN network again. This process continues for 3 recycling iterations and the end result undergoes a final "relaxation" step that uses physical restraint-based losses to optimize the positions of the model atoms using an L-BFGS optimizer (Liu & Nocedal, 1989). This step mainly alleviates unnatural side chain distance violations, and does not noticeably affect the distance metrics in section 4, which are all based on main chain atoms.

There are two types of output from the model. In the unpruned output, chains are built more or less as-is from the output of the model, with minimal post-processing. This allows for a larger percentage of the model to be built (higher recall), however this also means that portions of the model that exist in lower resolution areas of the map are often wrong. This can be cumbersome for biologists to use as they have to manually prune and fix chains that are incorrect and closely analyze the model. The pruned output seeks to remedy this by removing portions of the model that do not have good matches to the sequence based on the hidden Markov model alignment. This leads to lower recall, but higher sequence matching percentages, correlating to the correctly modelled portion of the map. The pruned output is often preferred by the biologists, who still have the option to analyze the sections that are lower confidence from the model in the unpruned output.

4 RESULTS

Here we report results for a test dataset of 28 map-model pairs that were deposited to the PDB and EMDB after the cutoff date for training. Generally, the atomic models built by ModelAngelo are close to the deposited PDB structures and they degrade with the resolution of the cryo-EM map.

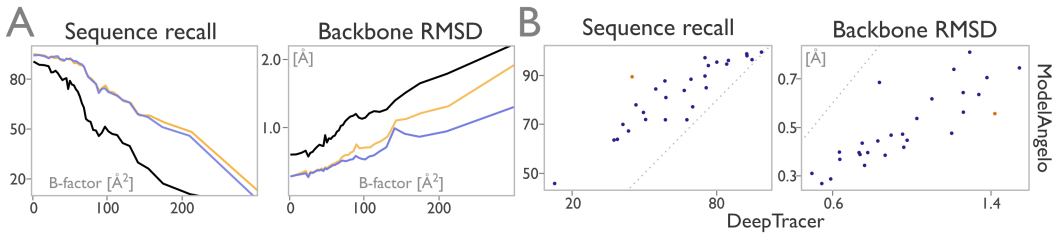

Figure 3: **(A)** shows sequence recall and backbone RMSD averaged for all residues in the test dataset as a function of B-factor labels, for Deeptracer (black) and ModelAngelo (before pruning in orange; after pruning in purple). **(B)** shows the same results, but averaged for each PDB entry, with ModelAngelo's pruned prediction (y-axis) versus Deeptracer (x-axis). The dotted line marks the identity line. The orange marker represents PDB entry 8DTM, which is shown in figure 4.

The overall resolution of the 28 test maps ranges from 2.1 to 3.8 Å. However, flexibility in parts of the protein structures also leads to local variations in resolution across the maps. The latter are reflected in the refined B-factors of the individual residues in the deposited PDB coordinate files, where higher B-factors indicate lower local resolution. We compare our results against the current state-of-the-art method for automated model building, Deeptracer (Pfab et al., 2021).

We consider two metrics. Sequence recall is the percentage of residues for which the C^{α} atom is within 3 Å of the deposited model, and the amino acid prediction is correct. Backbone RMSD is the root mean square deviation of all the backbone atoms in Ångstroms. We did not calculate all-atom RMSDs, as Deeptracer does not output side chain coordinates. Figure 3 compares the performance of ModelAngelo and Deeptracer for each of the structures in the test dataset, and as a function of B-factor averaged over all residues. A more comprehensive list of metrics, for each of the 28 map-model pairs, is shown in Appendix A.3. Over the entire test dataset, there is little difference in sequence recall between the unpruned and the pruned model from ModelAngelo, whereas the backbone RMSD does improve after pruning. This implies that pruning removes incorrectly built parts of the model. We believe that the improved results of ModelAngelo versus Deeptracer are because ModelAngelo is able to combine different modalities of information to build the model, rather than just the cryo-EM map. Figure 4 illustrates the quality of the built models from ModelAngelo's and Deeptracer for one example from the test dataset (PDB entry 8DTM). Because of its increased complexity, ModelAngelo is considerably slower than Deeptracer. Still, execution times for the test dataset are in the range of several minutes to one hour and a half, depending on the size of the structure. Given that manual *de novo* model building takes on the order of weeks, we do not believe this to be a serious drawback.

5 Discussion

Our results illustrate that ModelAngelo is a useful tool for automating the intensive task of atomic modelling in cryo-EM maps. Below, we consider limitations of the current implementation of ModelAngelo, and outline lines of future research to overcome them.

Sensitivity to resolution. Even though the network has multiple different modalities of input, relatively low resolutions of the cryo-EM map will affect the results. The initial C^{α} placement by the CNN, and the amino acid classification that provides the information for mapping the sequence onto the main chain, are obvious examples that benefit from higher resolution maps. Poor amino acid classifications may also lead to errors in the sequence assignment in the postprocessing step, which may then feed into the subsequent HMM sequence alignment and lead to incorrect chain assignments. This is more likely to happen for complexes with many similar sequences. In practice, we observe that ModelAngelo's performance starts degrading at resolutions worse than 3.5 Å, see Appendix A.3. It may be possible to combine the embedding of information from relatively low-resolution cryo-EM maps (e.g 10 Å) with methods for protein structure prediction, such as AlphaFold2 (Jumper et al., 2021). Despite the observation that ModelAngelo already uses some ideas from AlphaFold2, such an approach for building in low-resolution maps, which would blur

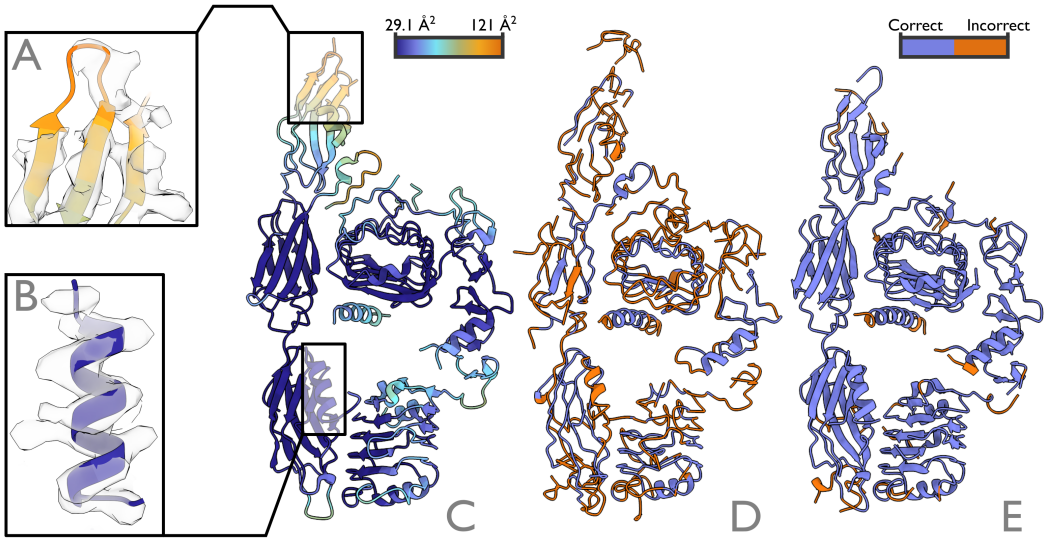

Figure 4: Comparison of the deposited model for PDB entry 8DTM **(C)**, Deeptracer's prediction **(D)** and ModelAngelo's pruned model **(E)**. The deposited model is coloured according to the refined B-factor. Meanwhile, the predictions are coloured orange where their amino acid prediction is different from the deposited structure, and purple where it is the same. **(A)** shows an iso-surface of the cryo-EM map and the deposited model for a high B-factor region, with correspondingly poor density. **(B)** shows the same for a low B-factor region, where side chains are well resolved in the density.

the boundaries between experimental structure determination and prediction, would require major changes to the approaches outlined in this paper.

Nucleic acids. Many large complexes that are solved with cryo-EM comprise nucleic acids as well as proteins, e.g. ribonucleic acids (RNA) in spliceosomes and ribosomes, or deoxyribonucleic acids (DNA) in gene replication or transcription machinery. The backbone of DNA or RNA strands is made from alternating phosphate and sugar groups. The phosphorus atoms has high contrast in the cryo-EM map, which makes the segmentation problem of nucleic acids easier than that of protein residues. However, the main difficulty lies in identifying the correct sequence for the nucleobases that make up the equivalent of side chains for RNA or DNA strands. There are four typical bases for both RNA and DNA: two purines and two pyrimidines. At resolutions around 3.5 Å, one can distinguish the purines from the pyrimidines, but not the two purines or pyrimidines from each other. This makes sequencing DNA or RNA strands in 3.5 Å cryo-EM maps difficult. Often, RNA or DNA sequence assignments are made based on additional sources of information, beyond the cryo-EM map itself. Therefore, extending ModelAngelo to include full atomic modelling of nucleic acids in such maps will be difficult. Instead, we plan to add support for building RNA or DNA backbones, together with a classifier to distinguish purines from pyrimidines. The resulting models would alleviate the task of manual model building, but assignment of the sequence would probably remain, at least partially, supervised.

Unknown sequences. Because cryo-EM can be performed on samples that are extracted from native cells or tissues, it is not always obvious which proteins are present in a cryo-EM map, (for an example, see Schweighauser et al., 2022). The current implementation of ModelAngelo depends on a user-provided sequence file that defines all proteins present in the map. Recently, semi-automated tools to identify proteins in cryo-EM maps have been reported. For example, findMySequence (Chojnowski et al., 2022) can search protein seqeuence databases, using amino acid classifications after a backbone model has been built in the cryo-EM map. We plan to extend ModelAngelo with a sequence-free model to fully automate this process, by combining automated model building with searches through large sequence databases like Uniclust (Mirdita et al., 2017) using our HMM search procedure combined with tools such as HHblits (Remmert et al., 2012).

REFERENCES

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M Duarte, et al. Rcsb protein data bank: powerful new tools for exploring 3d structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic acids research*, 49(D1):D437–D451, 2021.

Yifan Cheng. Single-particle cryo-em—how did it get here and where will it go. *Science*, 361(6405): 876–880, 2018.

Grzegorz Chojnowski, Adam J Simpkin, Diego A Leonardo, Wolfram Seifert-Davila, Dan E Vivas-Ruiz, Ronan M Keegan, and Daniel J Rigden. findmysequence: a neural-network-based approach for identification of unknown proteins in x-ray crystallography and cryo-em. *IUCrJ*, 9(1), 2022.

Kevin Cowtan. The buccaneer software for automated model building. 1. tracing protein chains. *Acta crystallographica section D: biological crystallography*, 62(9):1002–1011, 2006.

Tristan Ian Croll. Isolde: a physically realistic environment for model building into low-resolution electron-density maps. *Acta Crystallographica Section D: Structural Biology*, 74(6):519–530, 2018.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning based protein sequence design using proteinmpnn. *bioRxiv*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Paul Emsley, Bernhard Lohkamp, William G. Scott, and Kevin Cowtan. Features and development of coot. *Acta Crystallographica Section D - Biological Crystallography*, 66:486–501, 2010.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi S. Jaakkola, and Andreas Krause. Independent SE(3)-equivariant models for end-to-end rigid protein docking. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=GQjaI9mLet.

Jonathan Godwin, Michael Schaarschmidt, Alexander Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Velickovic, James Kirkpatrick, and Peter W. Battaglia. Very deep graph neural networks via noise regularisation. *CoRR*, abs/2106.07971, 2021. URL https://arxiv.org/abs/2106.07971.

Seyed Raein Hashemi, Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, Sanjay P. Prabhu, Simon K. Warfield, and Ali Gholipour. Tversky as a loss function for highly unbalanced image segmentation using 3d fully convolutional deep networks. *CoRR*, abs/1803.11078, 2018. URL http://arxiv.org/abs/1803.11078.

Jiahua He, Peicong Lin, Ji Chen, Hong Cao, and Sheng-You Huang. Model building of protein complexes from intermediate-resolution cryo-em maps with deep learning-guided automatic assembly. *Nature Communications*, 13(1):1–16, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Anders Krogh, Michael Brown, I Saira Mian, Kimmen Sjölander, and David Haussler. Hidden markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–1531, 1994.

Catherine L Lawson, Ardan Patwardhan, Matthew L Baker, Corey Hryc, Eduardo Sanz Garcia, Brian P Hudson, Ingvar Lagerstedt, Steven J Ludtke, Grigore Pintilie, Raul Sala, et al. Emdatabank unified data resource for 3dem. *Nucleic acids research*, 44(D1):D396–D403, 2016.

Dorothee Liebschner, Pavel V Afonine, Matthew L Baker, Gábor Bunkóczi, Vincent B Chen, Tristan I Croll, Bradley Hintze, L-W Hung, Swati Jain, Airlie J McCoy, et al. Macromolecular structure determination using x-rays, neutrons and electrons: recent developments in phenix. *Acta Crystallographica Section D: Structural Biology*, 75(10):861–877, 2019.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017b. URL `http://arxiv.org/abs/1708.02002`.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and Alexander Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022. doi: 10.1101/2022.07.20.500902. URL `https://www.biorxiv.org/content/early/2022/07/21/2022.07.20.500902`.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

Milot Mirdita, Lars Von Den Driesch, Clovis Galiez, Maria J Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2017.

Jaina Mistry, Robert D Finn, Sean R Eddy, Alex Bateman, and Marco Punta. Challenges in homology search: Hmmer3 and convergent evolution of coiled-coil regions. *Nucleic acids research*, 41 (12):e121–e121, 2013.

Garib N Murshudov, Pavol Skubák, Andrey A Lebedev, Navraj S Pannu, Roberto A Steiner, Robert A Nicholls, Martyn D Winn, Fei Long, and Alexei A Vagin. Refmac5 for the refinement of macromolecular crystal structures. *Acta Crystallographica Section D: Biological Crystallography*, 67(4):355–367, 2011.

Takanori Nakane, Abhay Kotecha, Andrija Sente, Greg McMullan, Simonas Masiulis, Patricia MGE Brown, Ioana T Grigoras, Lina Malinauskaite, Tomas Malinauskas, Jonas Miehling, et al. Single-particle cryo-em at atomic resolution. *Nature*, 587(7832):152–156, 2020.

Anastassis Perrakis, Richard Morris, and Victor S Lamzin. Automated protein model building combined with iterative structure refinement. *Nature structural biology*, 6(5):458–463, 1999.

Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Elaine C Meng, Gregory S Couch, Tristan I Croll, John H Morris, and Thomas E Ferrin. Ucsf chimerax: Structure visualization for researchers, educators, and developers. *Protein Science*, 30(1):70–82, 2021.

Jonas Pfab, Nhut Minh Phan, and Dong Si. Deeptracer for fast de novo cryo-em protein structure modeling and special studies on cov-related complexes. *Proceedings of the National Academy of Sciences of the United States of America*, 118, 1 2021. ISSN 10916490. doi: 10.1073/PNAS.2017525118/SUPPL_FILE/PNAS.2017525118.SAPP.PDF.

Ali Punjani, John L Rubinstein, David J Fleet, and Marcus A Brubaker. cryosparc: algorithms for rapid unsupervised cryo-em structure determination. *Nature methods*, 14(3):290–296, 2017.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021.

Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

James P. Roney and Sergey Ovchinnikov. State-of-the-art estimation of protein model accuracy using alphafold. *bioRxiv*, 2022. doi: 10.1101/2022.03.11.484043. URL https://www.biorxiv.org/content/early/2022/06/19/2022.03.11.484043.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Peter B Rosenthal and Richard Henderson. Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy. *Journal of molecular biology*, 333(4):721–745, 2003.

Sjors HW Scheres. Relion: implementation of a bayesian approach to cryo-em structure determination. *Journal of structural biology*, 180(3):519–530, 2012.

Manuel Schweighauser, Diana Arseni, Mehtap Bacioglu, Melissa Huang, Sofia Lövestam, Yang Shi, Yang Yang, Wenjuan Zhang, Abhay Kotecha, Holly J Garringer, et al. Age-dependent formation of tmem106b amyloid filaments in human brains. *Nature*, 605(7909):310–314, 2022.

Thomas C Terwilliger, Ralf W Grosse-Kunstleve, Pavel V Afonine, Nigel W Moriarty, Peter H Zwart, L-W Hung, Randy J Read, and Paul D Adams. Iterative model building, structure refinement and density modification with the phenix autobuild wizard. *Acta Crystallographica Section D: Biological Crystallography*, 64(1):61–69, 2008.

Thomas C. Terwilliger, Billy K. Poon, Pavel V. Afonine, Christopher J. Schlicksup, Tristan I. Croll, Claudia Millán, Jane. S. Richardson, Randy J. Read, and Paul D. Adams. Improved alphafold modeling with implicit experimental information. *bioRxiv*, 2022. doi: 10.1101/2022.01.07.475350. URL https://www.biorxiv.org/content/early/2022/01/30/2022.01.07.475350.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022a. doi: 10.1101/2022.07.21.500999. URL https://www.biorxiv.org/content/early/2022/07/22/2022.07.21.500999.

Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022b.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Ka Man Yip, Niels Fischer, Elham Paknia, Ashwin Chari, and Holger Stark. Atomic-resolution protein structure determination by cryo-em. *Nature*, 587(7832):157–161, 2020.

A Appendix

A.1 Losses

This is a full description of all the losses used for training the GNN. These losses can be divided into two groups: one-step losses and auxiliary losses. One-step losses are losses that are only defined once for the final step of the GNN. However, auxiliary losses are summed over each step of the GNN, though the last step has a higher weighting.

A.1.1 Auxiliary Losses

These losses are as follows: C^α RMSD loss, backbone frame loss, amino-acid classification loss, edge classification loss, local confidence score loss, and existence loss.

The **C^α RMSD loss** is just the RMSD of the C^α atoms only. That is

$$\begin{aligned}\mathcal{L}_{C^\alpha} &= \frac{1}{N}\sum_i \mathrm{RMSD}(\mathbf{x}_i, f_\theta \mathbf{x}_i + \mathbf{e}_i) \\ &= \frac{1}{N}\sum_i \sqrt{\frac{1}{3}\sum_{d=1}^{3}\left|[\mathbf{x}_i]_d - [f_\theta(\mathbf{x}_i + \mathbf{e}_i)]_d\right|^2}\end{aligned}$$

The **backbone frame loss** is the RMSD of the three backbone atoms that define the backbone frame: C^α, C, and N atoms. This means that the RMSD of the C^α is over-represented, which is what we want.

The **amino-acid classification loss** is the focal loss between the amino-acid classification logits and the target amino-acid recorded for each node. We have observed that even after convergence of the GNN, the auxiliary loss of this classification task is relatively high, meaning that earlier layers are not well capable of distinguishing amino acids. This could be because the C^α positions are still noisy or that the network needs to pool neighbouring information more before it can be confident about the classification.

The **edge classification loss** is based on whether two ground truth C^α nodes are within 3.9 Å of each other. If they are, then we expect both nodes a and b to classify each other as a neighbour. Otherwise, both need to classify the other as not a neighbour.

The **local confidence score loss** is a simple regression loss that tries to predict the backbone frame loss described above for each residue. This gives a good measure for how confident the network is in the position of the residue. This measure is then normalized between 0 and 100 based on a simple heuristic so that larger expected losses lead to lower confidence values.

Finally, the **existence loss** is a focal loss that classifies whether a residue is an artificially added residue or it actually exists.

A.1.2 ONE-STEP LOSSES

The **full-atom loss** is the same RMSD loss defined above for the C^α atom, except that it is calculated for all atoms, including side chain atoms.

The **torsion angles loss** is defined as the L_2 distance between the torsion angles of the network output and the torsion angles of the deposited model. A small loss is added to make sure the norm of the torsion angles is equal to 1. This is similar to the loss defined in section 1.9.1 of Jumper et al. (2021).

The **sequence attention loss** is a classification loss defined over the length of the given sequence for each residue, designed to match the position of the graph node. It is calculated for the last attention head of the sequence attention module as a focal loss over the calculated attention score.

The **violation loss** is meant to constrain the network to output models that are more in line with ideal bond lengths and angles. This loss is the same as defined in section 1.9.11 in Jumper et al. (2021).

A.2 SYMBOLS

Symbol Names

N	Number of residues/nodes
d	Dimension of feature vectors
v	Box size of the cryo-EM map
k	Number of neighbours, default set to 20
p	Number of query points per node, default set to 4
$\boldsymbol{V}$	cryo-EM map, $\mathbb{R}^{v\times v\times v}$
$\mathbf{x}$	C^α positions, $\mathbb{R}^{N\times 3}$
$\mathbf{z}$	Node features, $\mathbb{R}^{N\times d}$
$\mathbf{F}$	Backbone frames, $\mathbb{R}^{N\times 3\times 4}$
$[\cdot,\cdot]$	Concatenation, $\mathbb{R}^{\cdots a}, \mathbb{R}^{\cdots b} \rightarrow \mathbb{R}^{\cdots a+b}$

A.3 DETAILED METRICS PER PDB ENTRY

The tables below contain the resolution (in Å) and various metrics for each of the 28 PDB entries in the test dataset. Backbone RMSD and sequence recall are defined in the main text. C^α recall is the percentage of residues for which the C^α atom is within 3 Å of the deposited model; C^α precision is the percentage of C^α atoms predicted that are within 3 Å of a C^α atom in the deposited map; lDDT is the C^α local distance difference test as described in Mariani et al. (2013); and sequence match is the percentage of matching amino-acids for the corresponding residues. Table 1 describes the results obtained by Deeptracer; tables 2 and 3 describe the results obtained by ModelAngelo, after and before pruning, respectively.

PDB	Reso-lution	backbone RMSD	C^{α} recall	C^{α} precision	lDDT	Sequence match	Sequence recall
7tu5	2.1	0.591	94.3	99.4	96.7	90.0	84.8
7unl	2.45	0.496	97.3	98.2	97.4	95.0	92.4
7q1u	2.7	0.546	99.3	99.2	96.2	99.2	98.5
7szi	2.7	0.635	98.3	99.0	94.2	96.3	94.6
7txv	2.7	0.837	76.3	97.8	91.2	77.3	58.9
7uck	2.8	0.866	87.8	49.0	93.6	85.5	75.1
7ode	2.84	0.960	66.6	29.0	93.1	69.9	46.5
7sba	2.9	0.829	88.4	93.9	92.2	75.2	66.5
7ugg	3.16	0.896	93.0	98.7	93.2	82.3	76.5
7pt6	3.2	0.750	92.4	99.0	93.3	91.1	84.1
7xpx	3.2	0.762	82.0	58.8	96.1	92.2	75.6
7px8	3.27	0.637	97.6	99.2	94.0	94.7	92.5
7sr8	3.3	1.026	95.4	95.3	86.6	78.8	75.1
7wug	3.3	0.778	97.1	95.8	92.5	82.5	80.0
7y9u	3.3	0.739	82.9	98.7	94.3	84.5	70.0
7u50	3.4	0.734	76.6	88.3	92.9	88.3	67.6
7sjn	3.4	1.213	70.5	95.5	85.7	53.1	37.4
7z1m	3.4	0.954	92.9	97.7	89.2	74.6	69.3
7rzy	3.5	1.544	71.4	94.2	79.7	17.9	12.8
7w9l	3.5	0.977	80.6	96.1	90.2	72.8	58.7
8dtm	3.5	1.420	95.9	81.6	87.1	46.8	44.9
8a3t	3.5	1.380	76.5	96.6	81.1	50.8	38.8
8a2q	3.53	1.260	74.0	96.6	85.2	55.5	41.1
7yzk	3.57	1.293	94.8	88.4	82.7	53.2	50.5
7oix	3.6	1.339	95.1	77.9	82.0	45.7	43.4
7tvz	3.6	1.261	91.8	96.3	85.9	59.6	54.7
7pt7	3.8	1.202	86.3	96.4	86.5	57.5	49.6
7um0	3.8	1.102	86.2	98.1	86.7	67.8	58.5

Table 1: Deeptracer Results

PDB	Reso-lution	backbone RMSD	C^α recall	C^α precision	lDDT	Sequence match	Sequence recall
7tu5	2.1	0.287	96.3	99.7	98.7	99.9	96.2
7unl	2.45	0.310	99.2	94.7	98.9	99.7	98.9
7q1u	2.7	0.268	99.6	99.3	99.2	99.9	99.6
7szi	2.7	0.398	97.2	99.4	98.4	99.3	96.5
7txv	2.7	0.686	89.5	91.5	92.3	90.5	81.0
7uck	2.8	0.385	97.6	99.5	97.4	99.8	97.3
7ode	2.84	0.418	81.5	99.2	96.7	95.7	78.0
7sba	2.9	0.444	86.1	98.2	96.1	97.4	83.9
7ugg	3.16	0.468	96.3	98.6	95.9	97.7	94.1
7pt6	3.2	0.435	95.7	99.1	96.6	98.9	94.7
7xpx	3.2	0.343	85.4	97.1	97.6	99.5	85.0
7px8	3.27	0.368	99.0	98.2	97.7	99.1	98.1
7sr8	3.3	0.537	92.7	98.0	93.5	96.9	89.8
7wug	3.3	0.396	96.2	94.1	97.0	99.3	95.5
7y9u	3.3	0.388	77.9	99.0	97.7	99.1	77.2
7u50	3.4	0.396	73.7	97.7	97.0	97.6	71.9
7sjn	3.4	0.740	68.3	96.3	90.5	93.1	63.6
7z1m	3.4	0.472	90.4	99.3	95.3	97.9	88.5
7rzy	3.5	0.746	47.2	99.2	90.1	97.2	45.8
7w9l	3.5	0.447	74.0	98.7	96.2	97.2	71.9
8dtm	3.5	0.557	92.0	90.9	93.8	97.3	89.5
8a3t	3.5	0.706	66.9	98.6	90.4	95.6	63.9
8a2q	3.53	0.644	74.1	94.8	90.8	94.4	70.0
7yzk	3.57	0.811	81.0	94.4	88.5	88.9	72.0
7oix	3.6	0.636	75.1	96.4	91.2	89.6	67.3
7tvz	3.6	0.563	86.5	98.3	93.3	97.7	84.5
7pt7	3.8	0.476	78.2	99.5	95.0	95.9	74.9
7um0	3.8	0.618	89.8	99.0	92.2	97.7	87.8

Table 2: ModelAngelo Pruned Results

PDB	Resolution	backbone RMSD	C^α recall	C^α precision	lDDT	Sequence match	Sequence recall
7tu5	2.1	0.307	99.7	98.1	98.3	96.7	96.5
7unl	2.45	0.311	99.9	84.8	98.8	98.9	98.8
7q1u	2.7	0.273	100.0	97.7	99.1	99.6	99.6
7szi	2.7	0.406	99.1	98.2	98.3	97.8	96.9
7txv	2.7	0.710	92.3	87.0	91.9	88.4	81.6
7uck	2.8	0.401	99.0	98.2	97.2	98.4	97.4
7ode	2.84	0.549	96.9	93.5	94.6	84.2	81.7
7sba	2.9	0.516	97.4	91.5	94.7	87.6	85.3
7ugg	3.16	0.495	98.9	97.1	95.4	95.4	94.3
7pt6	3.2	0.466	98.1	97.7	96.1	96.6	94.9
7xpx	3.2	0.563	98.2	90.3	93.7	88.1	86.5
7px8	3.27	0.371	99.9	96.5	97.7	98.5	98.4
7sr8	3.3	0.574	99.0	93.7	92.7	92.0	91.1
7wug	3.3	0.420	99.8	82.0	96.5	96.1	95.9
7y9u	3.3	0.719	99.1	86.0	91.7	82.8	82.1
7u50	3.4	0.848	99.1	67.2	89.6	76.3	75.6
7sjn	3.4	1.071	98.6	78.7	85.8	70.2	69.2
7z1m	3.4	0.546	98.0	97.6	93.9	91.4	89.6
7rzy	3.5	1.122	95.3	85.9	82.8	57.2	54.5
7w9l	3.5	0.646	90.2	89.2	92.6	83.0	74.9
8dtm	3.5	0.649	99.5	74.8	92.1	90.5	90.0
8a3t	3.5	0.960	86.5	94.3	85.9	77.7	67.3
8a2q	3.53	0.923	98.3	82.7	86.6	74.7	73.5
7yzk	3.57	0.935	99.4	75.4	86.2	74.4	73.9
7oix	3.6	0.843	99.4	74.9	87.6	71.1	70.7
7tvz	3.6	0.661	98.5	93.2	91.5	87.8	86.5
7pt7	3.8	0.583	88.9	97.4	93.2	86.2	76.6
7um0	3.8	0.727	98.6	93.8	90.5	90.5	89.2

Table 3: ModelAngelo Unpruned Results

A.4 ALGORITHMS

Algorithm 1 CryoEM Attention Module

function CRYO_EM_ATTENTION($\mathbf{x}, \mathbf{z}, \boldsymbol{V}$)
  $\boldsymbol{C}$ = get_cubes_centered_on_nodes($\mathbf{x}, \boldsymbol{V}$)
  $\boldsymbol{R}$ = get_rectangles_between_nodes($\mathbf{x}, \boldsymbol{V}$)
  $\mathbf{d}$ = generate_distance_based_features($\mathbf{x}$)
  $\mathbf{n}_e = [\mathbf{z}, \mathbf{d}]$
  $\mathbf{k}$ = get_cryo_key_vectors($\boldsymbol{R}$)
  $\mathbf{q}, \mathbf{v}$ = get_cryo_query_value_vectors($\mathbf{n}_e$)
  $\mathbf{z}_c = \mathrm{softmax}(\mathbf{q}^T\mathbf{k}) \cdot \mathbf{v}$
  $\mathbf{z}_p$ = get_cryo_point_features($\boldsymbol{C}$)
  $\mathbf{z}'$ = project_back_to_original_dimension($[\mathbf{z}_c, \mathbf{z}_p]$)
  $\mathbf{z}$ = layer_norm($\mathbf{z} + \mathbf{z}'$)

Algorithm 2 Spatial IPA Module

function SPATIAL_IPA($\mathbf{x}, \mathbf{z}, \mathbf{F}$)
  $\mathbf{d}$ = generate_distance_based_features($\mathbf{x}$)
  $\mathbf{n}_e = [\mathbf{z}, \mathbf{d}]$
  $\mathbf{v}$ = get_value_vector($\mathbf{n}_e$)
  $\mathbf{q}$ = get_query_vector_in_local_frame($\mathbf{n}_e$)
  $\mathbf{q} = \mathbf{F} \circ \mathbf{q}$ ▷ Bring to global frame
  $\mathbf{x}_n$ = get_neighbour_positions($\mathbf{x}$) ▷ Shape: $N \times k \times 3$
  $\mathbf{z}' = \mathrm{softmax}(-\sum_{i \in [p]} \|\mathbf{q}_i - \mathbf{x}_n\|^2) \cdot \mathbf{v}$ ▷ Sum is over query points
  $\mathbf{z}'$ = project_back_to_original_dimension($\mathbf{z}'$)
  $\mathbf{z}$ = layer_norm($\mathbf{z} + \mathbf{z}'$)