STRUCTURAL BIOLOGY

ISSN 2059-7983

Completion of partial structures using Patterson maps with the *CrysFormer* machine-learning model

Tom Pan,[a] Evan Dramko,[a] Mitchell D. Miller,[b] Anastasios Kyrillidis[a,c]* and George N. Phillips Jr[b,d]*

[a]Department of Computer Science, Rice University, Houston, TX 77005, USA, [b]Department of BioSciences, Rice University, Houston, TX 77005, USA, [c]Ken Kennedy Institute, Rice University, Houston, TX 77005, USA, and [d]Department of Chemistry, Rice University, Houston, TX 77005, USA. *Correspondence e-mail: anastasios@rice.edu, georgep@rice.edu

Received 29 May 2025
Accepted 31 October 2025

Edited by D. Harrus, European Bioinformatics Institute, United Kingdom

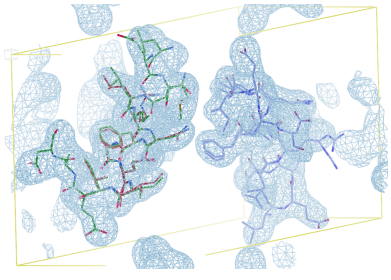
This article is part of the Proceedings of the CCP4 Study Weekend 2025.

Keywords: phasing; Patterson maps; machine learning; structure determination; X-ray crystallography.

Protein structure determination has long been one of the primary challenges of structural biology, to which deep machine learning (ML)-based approaches have increasingly been applied. However, these ML models generally do not directly incorporate the experimental measurements, such as X-ray crystallographic diffraction data. To this end, we explore an approach that more tightly couples these traditional crystallographic and recent ML-based methods by training a hybrid 3D vision transformer and convolutional network on inputs from both domains. We make use of two distinct input constructs: Patterson maps, which are directly obtainable from crystallographic data, and 'partial structure' template maps derived from predicted structures deposited in the AlphaFold Protein Structure Database with subsequently omitted residues. With these, we predict electron-density maps that are then post-processed into atomic models through standard crystallographic refinement processes. Introducing an initial data set of small protein fragments taken from Protein Data Bank entries and placing them in hypothetical crystal settings, we demonstrate that our method is effective at both improving the phases of the crystallographic structure factors and completing the regions missing from partial structure templates, as well as improving the agreement of the electron-density maps with the ground-truth atomic structures.

1. Introduction

Proteins are essential components of nearly all biochemical mechanisms performed in living cells (Tanford & Reynolds, 2004). They are composed of small organic molecules called amino acids (of which there are 20 typical proteinogenic ones) linked by peptide bonds; a single amino acid is often referred to as a residue. Protein functions are largely facilitated by their ability to bind only to specific molecules at specific sites on the protein, such that its 3D shape significantly informs its cellular activity. Thus, determining the characteristic complex 3D structure of a protein (which had folded from a polymer of amino-acid residues) is a longstanding problem of structural biology, having first been achieved by X-ray crystallography, and later by nuclear magnetic resonance (NMR) and cryo-electron microscopy (cryo-EM). All of these approaches face the problem of reconstructing an atomic structure given incomplete or imperfect experimental data (Drenth, 2007). In recent years, due to the wealth of high-quality information that has been accumulated in the Protein Data Bank (PDB) and vast sequence databases, machine learning has become another widespread approach for predicting protein structure, often with model architectures based on the transformer self-attention mechanism. Initiatives such as *AlphaFold*2 (Jumper



et al., 2021) and *AlphaFold*3 (Abramson *et al.*, 2024), which use sequence data in conjunction with co-evolutionary information in the form of multiple sequence alignments (MSAs), have demonstrated the ability of deep-learning models to produce variously precise atomic-level predictions. Other established ML-based approaches include *RoseTTAFold* (Baek *et al.*, 2021), *Boltz* (Wohlwend *et al.*, 2025) and *ESMFold* (Lin *et al.*, 2023), which does not depend on the creation of MSAs. However, certain issues remain (Terwilliger *et al.*, 2023), and X-ray crystallography is still frequently employed despite its well known associated difficulties (*i.e.* the crystallographic phase problem).

Thus, several projects have been developed with the purpose of bridging the gap between experimental crystallographic methods and ML techniques (see Matinyan *et al.*, 2024). In this work, we build upon our previous work in this area (Pan *et al.*, 2023, 2025) by developing an ML-based approach for improving predictions provided by other ML models, *i.e.* those in the AlphaFold Protein Structure Database (AFDB; Varadi *et al.*, 2024), given Patterson maps, which can be directly calculated from X-ray crystallography diffraction patterns without the need for phase information. We develop a novel synthetic data set of crystals of protein segments taken from PDB structures and corresponding AFDB entries, with a subset of residues subsequently omitted from the AFDB prediction templates, and show that a hybrid 3D vision transformer and convolutional neural network (CNN) can be trained to complete and improve the templates extracted from the *AlphaFold* predictions.

2. Problem setup and related work

2.1. X-ray crystallography

X-ray crystallography is one of the most commonly used experimental methods for determining the atomic-level structures of proteins and other large macromolecules (Lattman & Loll, 2008). In this technique, molecular crystals are exposed to X-ray beams, which diffract in specific directions according to the regular internal structure of the crystal to produce a pattern of spots (known as reflections).

Each reflection with a crystallographic diffraction pattern is associated with a set of three Miller indices h, k, l that indicate the orientation of sets of parallel planes within the crystal unit cell that contribute to producing the reflection (Ashcroft & Mermin, 2022), and can be shown to have an underlying representation known as a structure factor. Formally, a structure factor is the Fourier transformation of the electron density within the unit cell (the smallest repeating unit within a crystal). However, it can be well approximated as a discrete Fourier transform dependent on the atoms present in the crystal unit cell,

$$F(h,k,l)=\sum_{j=1}^{n} f_j \cdot \exp[2\pi i(hx_j+ky_j+lz_j)] \cdot \exp\left[-\frac{B_j}{4}\left(\frac{1}{d_{hkl}}\right)^2\right], \quad (1)$$

where f_j refers to the scattering factor property, B_j refers to the crystallographic B factor and (x_j, y_j, z_j) refers to the fractional coordinates of the jth atom within the cell (and all occupancies are assumed to be 1.0). Each of these structure factors is known to be a complex number, with both an amplitude and a phase (denoted by ϕ) component. An inverse Fourier transform can be taken over all such reflection structure factors to obtain an initial estimate of the electron density ρ at all points (x, y, z) within the unit cell, as

$$\rho(x,y,z)=\frac{1}{V}\cdot\sum_{h,k,l}|F(h,k,l)| \cdot \exp\{-2\pi i[hx+ky+lz-\phi(h,k,l)]\}, \quad (2)$$

where V is the volume of the unit cell. Once such a map of the electron density within the unit cell has been estimated, it is used to produce an initial input into iterative refinement programs, which perform repeated comparisons of the expected diffraction pattern given the current estimated model with experimental measurements, and eventually output a final atomic model. However, while the amplitude $|F(h, k, l)|$ of any reflection's underlying structure factor is simply proportional to the square root of its measured intensity, corresponding phase information cannot be immediately calculated from experimental crystallographic data. This is known as the crystallographic phase problem (Lattman & Loll, 2008).

2.2. Previous work for solving the crystallographic phase problem

Traditionally, three of the most widely used methods for addressing the crystallographic phase problem have been isomorphous replacement (IR), anomalous dispersion (AD) and molecular replacement (MR) (Lattman & Loll, 2008; Jin *et al.*, 2020). IR and AD almost always require multiple experimental settings, often with the production of molecular crystals with heavy-atom substitutions. On the other hand, MR requires the availability of homologous structures known to be similar to the current desired structure to be used as an initial template or phase estimate after a rotational and then translational search. Predictions made by the *AlphaFold*2 machine-learning model (Jumper *et al.*, 2021) have effectively been used as initial models for the MR technique (McCoy *et al.*, 2022), especially as part of an iterative process akin to traditional crystallographic refinement involving multiple rounds of MR and model building (Terwilliger *et al.*, 2023). Furthermore, in our previous work (Pan *et al.*, 2024), we obtained essentially *de novo* structural predictions of short protein fragments from corresponding Patterson maps using the *CrysFormer* model architecture. Our current work now aims to show the viability of incorporating an additional machine-learning step that makes use of our established *CrysFormer* model to improve existing *AlphaFold*2 predictions given crystallographic data, representing a further integration of experimental and ML-based protein structure-determination methods.

2.3. The Patterson function

The Patterson function (Patterson, 1934) is often used as an intermediary during the aforementioned methods for solving the crystallographic phase problem. It is a variation of the Fourier transform from structure factors to electron density where the amplitude components are squared and phases are ignored, resulting in what is called a Patterson map,

$$p(u, v, w) = \frac{1}{V} \cdot \sum_{h,k,l} |F(h, k, l)|^2 \cdot \exp[-2\pi i(hu + kv + lw)], \tag{3}$$

where (u, v, w) refers to locations within the Patterson map's unit cell, which has the same dimensions as that of the original crystal. Since phase information is not needed, Patterson maps can be immediately computed from raw crystallographic experimental data. If an underlying (real-valued, real-space) electron-density map is denoted as $\mathbf{e} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, then the corresponding Patterson map $\mathbf{p}$ can alternatively be formulated as

$$\mathbf{p} = \Re\{\mathcal{F}^{-1}[\mathcal{F}(\mathbf{e}) \odot \mathcal{F}(\widehat{\mathbf{e}})]\} = \Re\{\mathcal{F}^{-1}[|\mathcal{F}(\mathbf{e})|^2]\}, \tag{4}$$

where $\odot$ refers to element-wise multiplication, $\mathcal{F}$ refers to the Fourier transform and $\Re$ emphasizes that the result is a real number. $\widehat{\mathbf{e}}$ refers to an inverse-shifted version of $\mathbf{e}$, where each entry is defined as $\widehat{e}_{i,j,k} = e_{N_1-i,N_2-j,N_3-k}$.

From the construction of the Patterson function, a further derivation indicates that a Patterson map does not directly reveal the atomic structure within a unit cell. Instead (disregarding thermal motion effects and assuming infinite resolution), each peak in a Patterson map essentially corresponds to an interatomic vector between atoms within the crystal unit cell, and so Patterson maps of large macromolecules such as proteins are extremely dense with peaks (the amount of which scales quadratically with the number of atoms in the original cell) that may often blur together. Also, the height of these peaks is proportional to the product of atomic numbers in the corresponding pair (or the sum of all such pairs that have identical interatomic vectors), allowing the contributions of heavier atoms to dominate the resulting map. This is actually desired if they were explicitly incorporated or substituted into the molecular structure as in IR or AD. These issues prevent the straightforward interpretation of crystallographic Patterson maps, and so they have not been used to directly estimate the corresponding electron densities.

3. Model completion with partial structure inputs

3.1. Using deep learning

Patterson maps can be used as inputs into machine-learning models as constructs directly obtainable from raw crystallographic data without additional experiments or outside information. Thus, our goal is to train a model g with parameters $\boldsymbol{\theta}$ to estimate electron-density maps given corresponding Patterson maps as input (see Section 3.3). Formally, given a data set of n examples of the form $(\mathbf{p}_i, \mathbf{e}_i)_{i=1}^n$, where $\mathbf{p}_i \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is the Patterson map that corresponds to a ground-truth electron-density map, $\mathbf{e}_i \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, we aim to obtain optimal model parameters $\boldsymbol{\theta}^\star$ such that our model predictions are as close as possible to the ground-truth maps given a loss function $\mathcal{L}(\boldsymbol{\theta})$:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \mathcal{L}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell[g(\boldsymbol{\theta}, \mathbf{p}_i), \mathbf{e}_i] \right\}. \tag{5}$$

We use mean-squared error (MSE), which is well established for regression tasks, as our primary internal loss function. However, we also take the negative Pearson correlation between ground-truth and predicted maps as an additional loss-function term. This comparison between two constructs of the same shape is used across a wide range of application domains, including crystallography. Denoting a model prediction as $\mathbf{e}'$, and defining the average value over a ground-truth map and predicted map as $\bar{\mathbf{e}} = (1/N_1N_2N_3) \sum_{i,j,k} \mathbf{e}_{i,j,k}$ and $\bar{\mathbf{e}}' = (1/N_1N_2N_3) \sum_{i,j,k} \mathbf{e}'_{i,j,k}$, respectively, the Pearson correlation coefficient is defined as

$$\mathrm{PC}(\mathbf{e}, \mathbf{e}') = \frac{\sum_{i,j,k=1}^{N_1,N_2,N_3} (\mathbf{e}'_{i,j,k} - \bar{\mathbf{e}}')(\mathbf{e}_{i,j,k} - \bar{\mathbf{e}})}{\left[\sum_{i,j,k=1}^{N_1,N_2,N_3} (\mathbf{e}'_{i,j,k} - \bar{\mathbf{e}}')^2\right]^{1/2} \cdot \left[\sum_{i,j,k=1}^{N_1,N_2,N_3} (\mathbf{e}_{i,j,k} - \bar{\mathbf{e}})^2\right]^{1/2}}. \tag{6}$$

As larger Pearson correlations indicate greater agreement between maps, we negate the calculated values to incorporate them into our overall loss function.

3.2. Using existing predictions as partial structure templates

In our previous work (Pan *et al.*, 2024), the only additional input information provided to the model beyond Patterson maps was in the form of electron-density maps corresponding to single amino-acid residues in their most common conformations, which were referred to as 'partial structures'. However, for the current problem, we instead make use of existing predictions obtained from the AFDB as our partial structures. We omit a subset of residues from these existing predictions to simulate realistic conditions, where often portions of a protein structure prediction from an ML model would have regions of low confidence and accuracy (which we aim to fill in using experimental crystallographic data). We train our model to complete and improve these initial templates provided by the incomplete *AlphaFold* predictions, and thus no longer directly determine protein structures solely from crystallographic data, but instead incorporate both existing experimental and machine-learning approaches for structural prediction into a unified framework. We now aim to optimize

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \mathcal{L}(\boldsymbol{\theta}) := \frac{1}{n * J} \sum_{i=1}^{n} \sum_{j=1}^{J} \ell[\boldsymbol{\theta}; g, (\mathbf{p}_i, \mathbf{e}_i, \mathbf{u}_i^j)] \right\}. \tag{7}$$

where each original Patterson map and ground-truth pair $(\mathbf{p}_i, \mathbf{e}_i)$ is associated with multiple different corresponding incomplete template 'partial structures' $\mathbf{u}_i^j$, with each of these

(in practice up to) J partial structures having a different subset of removed residues. The full data-set size is denoted as $n*J$ in the equation for simplicity, but in practice is slightly smaller than this.

3.3. Model architecture

As stated, we continue to use the *CrysFormer* (Pan *et al.*, 2024) model introduced in our previous work for the model-completion task. This model is a hybrid of a 3D vision transformer and CNN, with Nyström approximate attention (Xiong *et al.*, 2021) in the self-attention layers of the transformer. For this work, we begin to use the scale-equivariant 3D convolution and batch-normalization layers introduced by Wimmer *et al.* (2023) in all such layers before the transformer. Also, we no longer provide several partial structure templates for each data-set example, each of a smaller size than the corresponding Patterson map input, but instead provide one single partial structure of the exact same size as the Patterson and desired ground-truth maps. Furthermore, every Patterson and ground-truth pair now corresponds to up to J distinct data-set examples (Fig. 1).

4. Data-set generation

We followed the same overall data-generation process as described in our previous work (Pan *et al.*, 2024), but with several modifications to better fit the new task of model completion on a single input partial structure (Fig. 2). We started with an expanded initial basis of nearly 38 000 PDB protein structures, curated according to the following criteria kept as before: solved by X-ray crystallography between the years 1995 and 2023, with sequence length ≥ 40, refinement resolution ≤ 2.75 Å, $R_{\rm free} \leq 0.28$ and available in legacy PDB format. As we desired a much larger data set for the current problem than that used in our previously reported work, we increased the clustering sequence-identity criterion from 30% to 70% to increase the size of the starting basis, and extracted all possible 15-residue fragments without randomly removing any obtained ones. Further, we allowed overlaps of up to five out of 15 residues between consecutive extracted fragments. Thus, we continued to place all examples derived from the same initial protein structure together in either the training set or the test set, preventing our model from potentially simply memorizing regions of protein segments that are present in the training set when evaluating unseen examples.

We applied most of our previous standardized modifications to these protein fragments, such as removing examples containing nonstandard or missing residues or missing atoms using the pdbfixer Python library (Eastman *et al.*, 2017), removing all H atoms and converting selenomethionine residues to methionine. As an additional form of variability, we did not reset all atomic temperature factors to a constant value, but instead kept all such values from the original PDB structure. We determined the original unit-cell extents for our fragments starting from the raw max–min ranges of Cartesian coordinates along each of the three axes, iteratively increasing the current dimensions until the minimum intermolecular

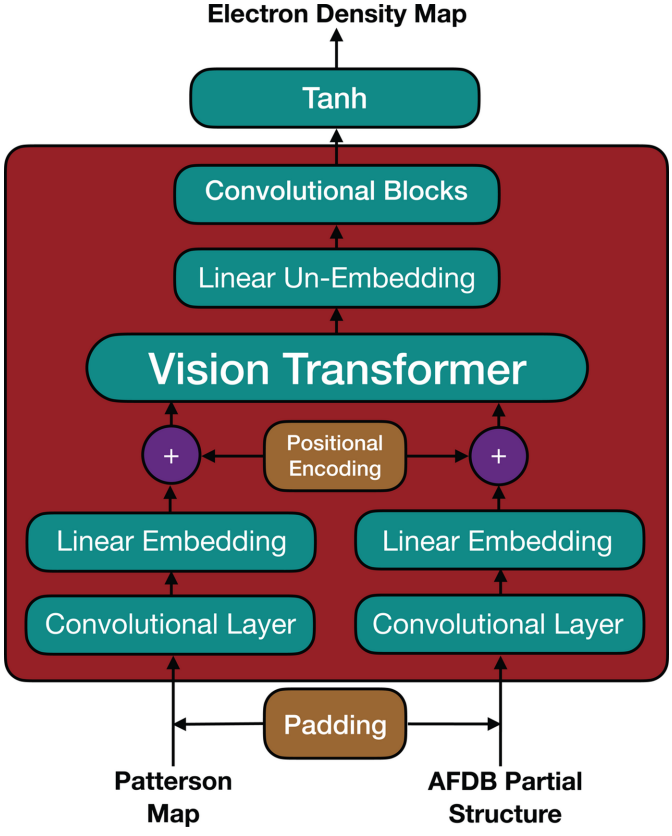


Figure 1
Overview of the *CrysFormer* model architecture.

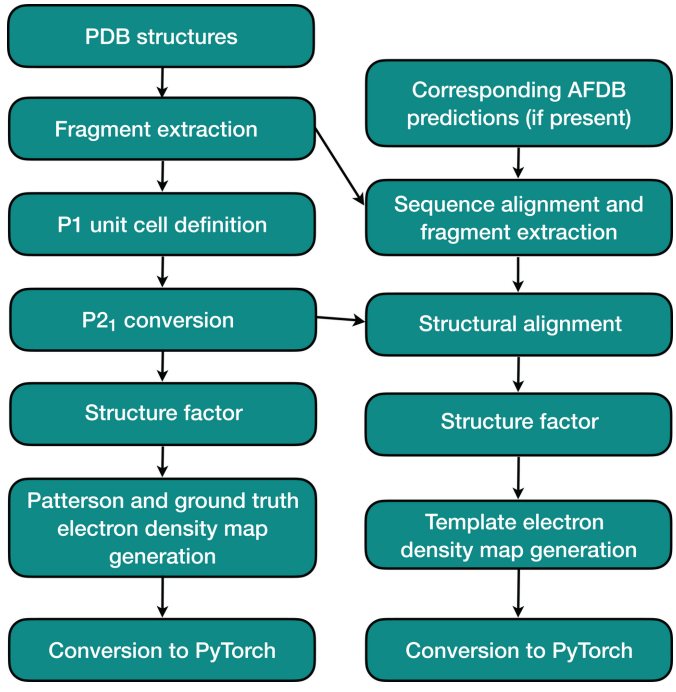


Figure 2
High-level steps of our data-set generation process.

atomic contact was at least 3.5 Å. We converted our examples to space group $P2_1$, which is one of the most common space groups (such unit cells contain two molecules related by a screw axis). First, we reoriented all initial unit cells so that the first axis is the longest and the second axis (along which the $P2_1$ screw axis is located by convention) is the shortest. Then, when converting to $P2_1$, we added an additional ångström to each original dimension and further expanded the length of the second axis by a multiplier randomly selected from the range 1.7–1.95. Another key consideration for this data set was obtaining a much more realistic solvent content (and thus the amount of empty space) within the unit cell compared with our previous data set, so we did not check for and remove examples that no longer satisfied the minimum 3.5 Å atomic contact requirement (Fig. 3). However, for each example, we still centered atomic coordinates such that the center of mass was at the exact center of the unit cell to avoid ambiguities associated with the translation invariance of Patterson maps; this is theoretically justifiable as unit-cell boundaries relative to the contents thereof are essentially arbitrary.

We again generated structure factors for each example in its final $P2_1$ unit cell with the *gemmi sfcalc* program (Wojdyr, 2022), without any bulk-solvent scaling or correction, and then created both Patterson and ground-truth electron-density maps in `.ccp4` format from these structure factors with the *FFT* program of the *CCP*4 program suite (Agirre *et al.*, 2023). We divided the data set as evenly as possible into 20 bins, each assigned to a different resolution limit in the range 1.75–2.3 Å, and restricted the structure factors used to generate the maps to the corresponding specified resolution limit. We also associated each of the resolution limit bins with a different grid-sampling factor in the range 2.29–2.7. We divided this value by the corresponding selected resolution limit, and used the result as a multiplier on the $P2_1$ unit-cell dimensions when specifying the map dimensions. As before, the Patterson and electron-density maps for each example had the exact same dimensions and resolution limits. The Patterson and ground-truth electron-density maps were then converted into PyTorch tensor format, maintaining the map axis dimensions, and all values in each tensor were normalized according to the maximum and minimum element values for the corresponding map type over the entire data set to be in the range $[-1, 1]$. The maximum and minimum values were found separately for the set of Patterson and electron-density maps, as Patterson maps are far more dense than electron densities.

To create our partial structure templates, we first queried the PDBe SIFTS database (Armstrong *et al.*, 2019; Dana *et al.*, 2019) to associate UniProt IDs with each of our original protein structures, given their PDB and entity IDs. For all structures with an associated UniProt ID, we obtained the associated AlphaFold ID if present in the `accession_ids.csv` file provided in the full AlphaFold Database (AFDB). If such an AlphaFold ID was found, we then downloaded the v4 version of the corresponding *AlphaFold* prediction from the AFDB (Varadi *et al.*, 2024). Structures without an AFDB match were excluded from the data set to avoid the computational overhead of running *AlphaFold* to generate suitable predicted models. We used the Needleman–Wunsch sequence-alignment method (Needleman & Wunsch, 1970), implemented as a Java program (Zhang & Yan, 2010), to align each of our protein-fragment examples with the *AlphaFold* prediction corresponding to the original structure it was extracted from. Using these alignments, we extracted all segments corresponding to our examples from the

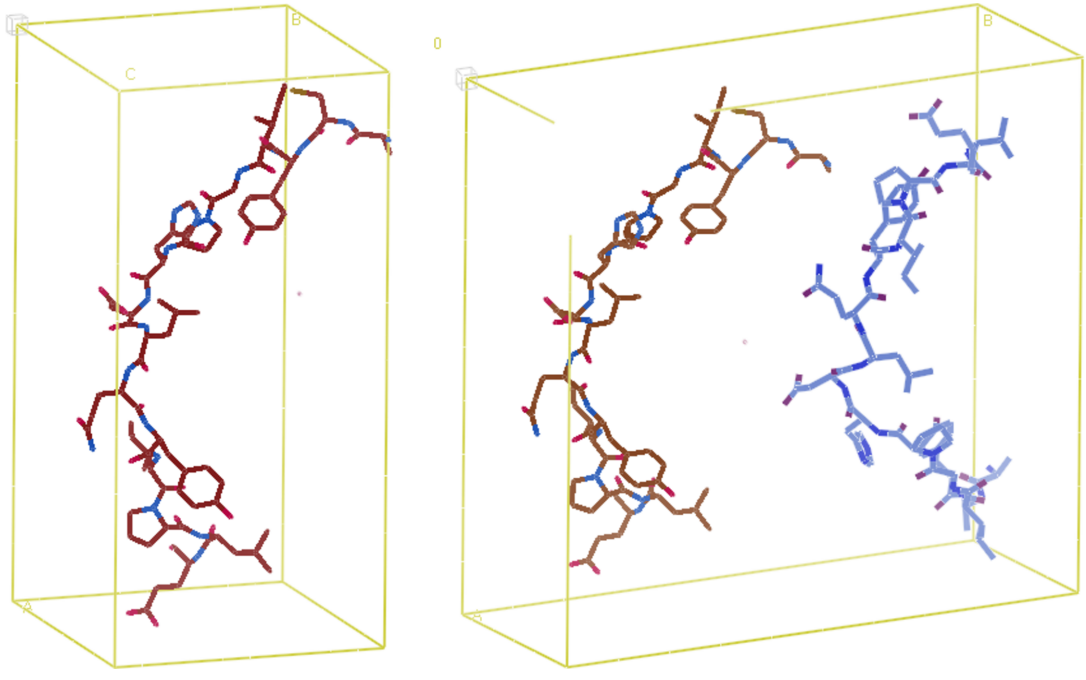


Figure 3
A 15-residue protein fragment extracted from a PDB structure in a reoriented $P1$ unit cell (left) and after conversion to $P2_1$ (right).

corresponding *AlphaFold* prediction coordinate files. After obtaining *AlphaFold* fragments for each remaining example, we applied the same set of standardized modifications as we had to the original fragments, although we reset all temperature factors for the atoms in the *AlphaFold* structures to a constant 20.0 Å^2. We then performed a structural alignment of each *AlphaFold* segment with its corresponding ground-truth segment. For computational expediency in generating the data set, we used the `align` command in *PyMOL* (version 2.5.5, Schrödinger; matching the backbone and C$^\beta$ atoms) as a proxy for fragment placement with an MR program (such as *Phaser*; McCoy *et al.*, 2007). We saved the aligned *AlphaFold* segment in a unit cell that matched the corresponding ground-truth coordinate file.

We then removed a subset of the residues from the *AlphaFold* fragments via a sequence of 2–3 random selections (Fig. 4). For each fragment, we generated up to three such partial structures with omitted residues. When removing residues, we first randomly selected whether to begin from the start of the coordinate file, the end of the file or both ends. Then, we randomly selected the number of residues to omit from three to seven out of 15 in total. If only one end to omit from was selected, we removed the selected number of residues contiguously. If both ends were selected, we omitted half of the total selected number from one end and the rest from the other end (if an odd number of residues was to be omitted, we simply performed another random selection to determine which end to remove more residues from). It is possible that the exact same end and number of omitted residues was selected as that of a previously created partial structure for a particular fragment. If this occurred, we did not try to create another partial structure, but simply allowed for fewer than J partial structures to be associated with that fragment (thus the total data-set size was slightly smaller than $n*J$). Afterwards, we applied the rest of our data-set generation process to the partial structures, but only needed to generate electron-density maps and not Patterson maps as well. For each partial structure, we specified the same map dimensions and resolution limit as the corresponding ground-truth fragment. When performing max–min normalization on the partial structure tensors, we used the exact same maximum and minimum as we had when normalizing the ground-truth electron-density tensors. Finally, to ensure uniform shape across the examples in each of our training batches as required by PyTorch, examples that belonged to tensor-size bins smaller than our minimum batch size of 6 were again excluded from the training set. This was far less likely than before as every ground-truth map in the training set is now associated with up to three distinct partial structures.

Table 1
Hyperparameter settings used in the training run.

Hyperparameter	Value
Convolution output channels	10
Patch size	4 × 4 × 4
Embedding dimension	512
Head dimension	64
No. of heads	12
MLP dimension	2048
Transformer layers	12
AdamW weight decay	3×10^{-2}
Inital lr	4.5×10^{-4}
Maximum lr	2.85×10^{-3}
Final lr	8.57×10^{-4}

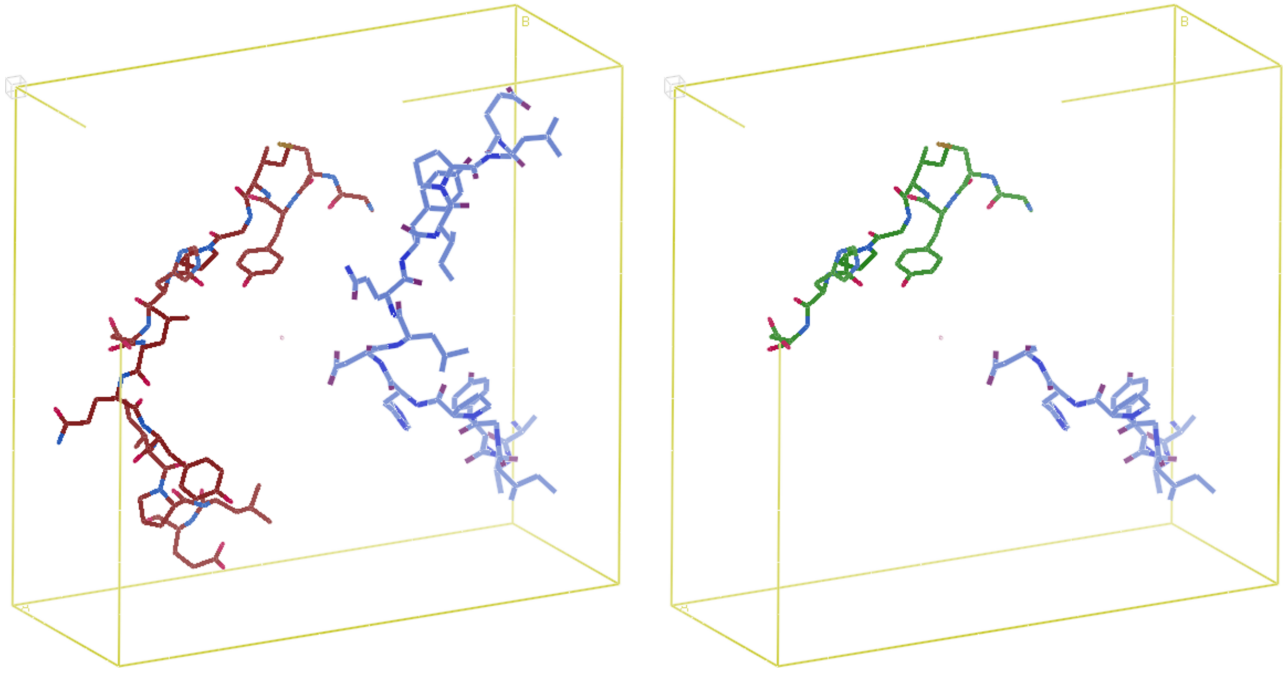


Figure 4
A 15-residue protein fragment extracted from a predicted structure in AFDB placed in a $P2_1$ unit cell (left) and a partial structure with a random amount of residues omitted from a random end of the extracted fragment (right).

Table 2
Comparison of structure factors derived from model predictions versus structure factors of partial structure templates after applying *SIGMAA*.

Metrics are averaged over test-set examples or a subset thereof. Standard deviations are reported in parentheses after mean values.

Metric	Predictions (Full)	*SIGMAA* (Full)	Predictions (Subset)	*SIGMAA* (Subset)
Phase error (unweighted)	**39.4** (10.3)	48.7 (10.5)	**52.7** (14.4)	62.6 (13.3)
Phase error (unweighted; cosine)	**0.762** (0.130)	0.649 (0.146)	**0.589** (0.209)	0.448 (0.205)
Phase error (weighted)	**30.2** (8.4)	38.5 (10.0)	**41.2** (13.4)	52.2 (14.7)
Phase error (weighted; cosine)	**0.856** (0.093)	0.772 (0.127)	**0.734** (0.177)	0.594 (0.212)
Pearson correlation coefficient	**0.867** (0.074)	0.816 (0.100)	**0.783** (0.136)	0.692 (0.171)

5. Experiments

5.1. Training details

We performed a single training run of our model on a training set of 589 546 initial Patterson map–ground truth electron-density pairs, where each original example was associated with up to three distinct 'partial structures' with 3–7 subsequently omitted residues out of 15 as described above ($J = 3$), for a total size of 1 634 839 examples. These were split into batches of minimum size 6, average size 10 and maximum size 11. For training, we used a 'Schedule-Free' variant of the AdamW optimizer (Defazio *et al.*, 2024), although we still enforced an overall OneCycle learning-rate schedule (Smith & Topin, 2019). The model was trained for 71 epochs in a data-parallel fashion using the DDP module of PyTorch (Li *et al.*, 2020) on a pair of RTX 6000 Ada GPUs with 48 GB memory each, with `torch.set_float32_matmul_precision` set to `high` and gradient accumulation performed every two batches. We report the hyperparameters of our model architecture used for this training run in Table 1. Furthermore, we downsample our structures by the patch size of 4 in every axis after the second and fourth transformer layers with a 3D convolution, and upsample by the same amount after the eighth and tenth transformer layers with a 3D transposed convolution.

5.2. Metrics

As a baseline comparison for our model predictions after training, we use the *SIGMAA* program from the *CCP*4 program suite (Agirre *et al.*, 2023) with the 'PARTIAL' option specified to improve the maps derived from partial structures with removed residues. Resolution ranges were specified to be the same as those used to generate `.ccp4` maps from structure factors during our data-set generation process. For evaluation, we use the *get_cc_mtz_pdb* program from the *Phenix* program suite (Liebschner *et al.*, 2019) to calculate the Pearson correlation coefficient, as defined previously, passing either post-*SIGMAA* structure factors or model prediction-derived structure factors without *SIGMAA* weighting, and the corresponding ground-truth atomic coordinates. We report the Pearson correlations calculated only over the region of the unit cell where the atomic model is located, ignoring empty regions. Additionally, we perform phase-error analysis using the *CPHASEMATCH* program (Cowtan, 2011), again from the *CCP*4 program suite. We report both unweighted and FOM-weighted average phase errors in degrees, where a smaller phase error is desirable, in Table 2.

5.3. Results and comparison with baseline

We provide a comparison of the predictions made by our model with the results after applying *SIGMAA* to the corresponding incomplete partial structure templates on our test examples in Table 2. Unweighted phase error considers the phase-error contribution of all structure factors equally for each example, while weighted phase error weighs the individual phase errors according to the associated figure of merit reported by *SIGMAA*. To obtain such weighted phase errors for our model predictions, we also applied *SIGMAA* with the same corresponding input parameters as described previously. The full test set consisted of 64 070 initial examples, once again with each associated by up to three partial structures with 3–7 omitted residues, for a total size of 176 556 examples. The reported subset consists of 19 436 (about 11%) test-set examples with the worst-performing structural alignments of *AlphaFold*2-derived partial structure to ground truth according to the root-mean-square deviation (r.m.s.d.) across all C^{α} atoms in the atomic structure. These examples had such C^{α} r.m.s.d.s ranging from 0.56 to 12.0 Å, while the full test set had a median r.m.s.d. of 0.22 Å.

Overall, the model predictions (see columns 1 and 3) show both noticeable improvement and (almost always) decreased variability on all metrics compared with the post-*SIGMAA* baseline (columns 2 and 4) on both the full test set (leftmost two columns) and the subset of examples for which the initial alignment was relatively poor (rightmost two columns).

We visualize some predictions in Fig. 5, with the first row consisting of more typical test-set examples and the second consisting of those that belong to the subset of test-set examples with the worst partial structure alignment C^{α} r.m.s.d. These figures further indicate that not only is the model effective at completing the portions missing from the partial structure template map, but it is also often able to improve regions of poor agreement between the partial structure and the true underlying ground truth.

We also aim to determine what effect, if any, the solvent content of an example has on its difficulty of completion, and so provide a plot of the in-model region Pearson correlations calculated by *get_cc_mtz_pdb* for our test-set model prediction structure factors against the solvent content of the corresponding ground truth in Fig. 6. It is clear from the distribution of Pearson correlations that the solvent content of

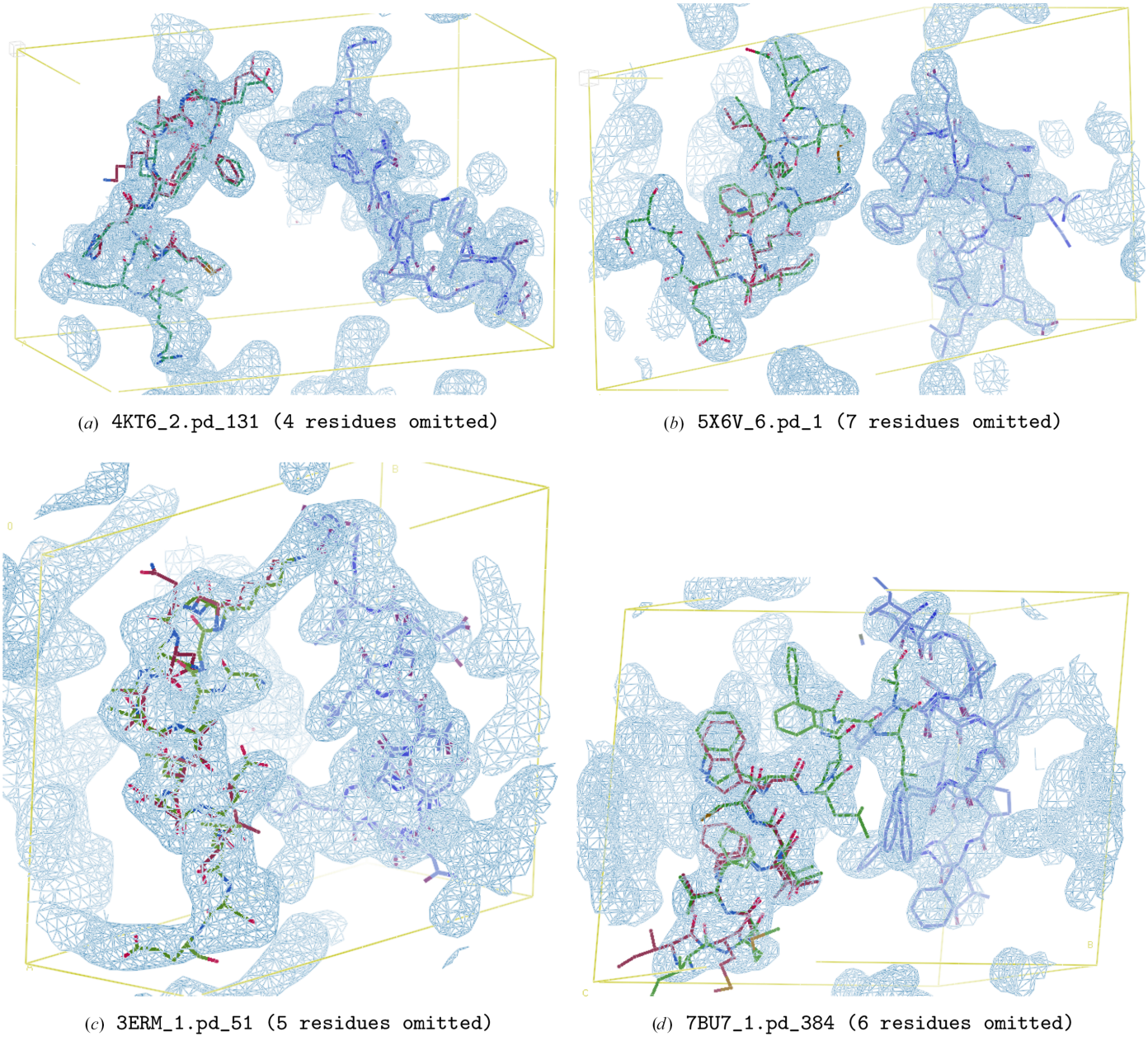
the ground-truth unit cell was not an underlying factor in how well the model was able to complete (and fix inaccurate regions of) a partial structure template.

6. Discussion

This work represents a key step in the integration of existing experimental and deep learning-based approaches for protein structure determination. We have established that our model, *CrysFormer*, can effectively 'complete' small protein fragments with omitted resides extracted from real predictions taken from the AlphaFold Database, when placed in unit cells corresponding to the desired ground truth and used as partial structure template maps alongside Patterson maps obtained from crystallographic data. Furthermore, the model is often able to greatly improve regions where the template maps were inaccurate.

6.1. Limitations and future work

The examples in the data set that we introduce in this work, although closer to true proteins than before, still fall short in realism in several aspects. Although each example has 30 residues divided across two molecules, this is still an unusually



(*a*) `4KT6_2.pd_131 (4 residues omitted)`

(*b*) `5X6V_6.pd_1 (7 residues omitted)`

(*c*) `3ERM_1.pd_51 (5 residues omitted)`

(*d*) `7BU7_1.pd_384 (6 residues omitted)`

Figure 5
Example visualizations of electron-density map predictions, shown in blue. The ground-truth atomic model is shown in green stick representation, while the partial structure atomic model used to generate the corresponding input template map is shown in red stick representation.

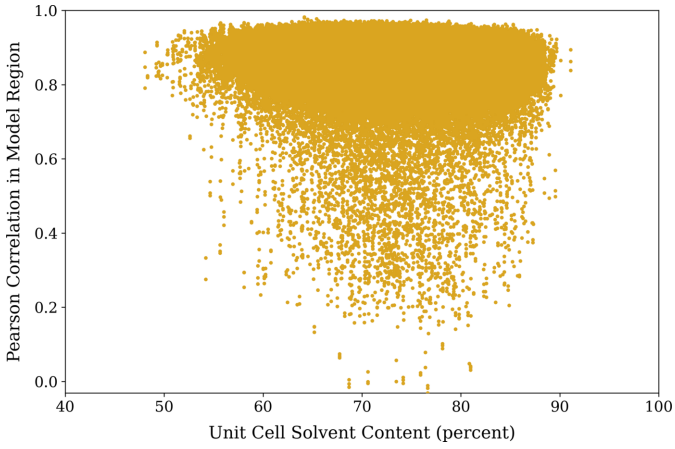


Figure 6
Pearson correlations of model predictions with ground truth versus underlying solvent content.

small number of residues per unit cell, and our examples still have an unusually high solvent percentage (and thus amount of empty space) in the unit cell. Thus, we are developing a new data set of examples with fragments consisting of entire protein domains of 50–150 residues in each asymmetric unit. This introduces another form of variability in our examples, as there will no longer be a constant number of residues in the unit cell across all examples. We will also use a wider potential range of fractions of residues removed in the training set to further increase model robustness.

Additionally, we want our model to handle more than just one type of internal symmetry at once. Thus, our new data set will contain examples belonging to one of five possible space groups, with up to four molecules per unit cell. We will provide multiple choices of space group and thus unit cell for each original domain example as yet another form of data augmentation. This will be necessary to maintain training-set size as we will be starting from a much smaller initial set of possible domain examples compared with 15-residue fragments. We have also not yet considered the effects of bulk solvent or experimental noise when generating our synthetic data examples.

Furthermore, the predictions made by our model after the training run can be incorporated into further ML methods and pipelines. A prediction can be used as an additional template map input for a subsequent 'recycling' iteration of training our model architecture (Pan *et al.*, 2025), or its derived structure factors can be used as a set of initial phase estimates for reciprocal-space phasing methods such as phase seeding (Carrozzini *et al.*, 2025).

Data availability

A repository, https://github.com/sciadopitys/CrysFormer_model_completion, contains our *CrysFormer* model architecture and training and batch-generation scripts. It also includes a subset of our data-set generation scripts to generate our Patterson, ground-truth and partial structure maps. Intermediate atomic coordinate files for fragments extracted from the PDB, sufficient to generate ground-truth files for the test and training sets, can be downloaded from https://doi.org/10.5281/zenodo.15498745. Files containing AFDB-derived fragments for partial structure generation can be downloaded from https://doi.org/10.5281/zenodo.15498821.

Funding information

This research was funded in part by The Robert A. Welch Foundation (grant No. C-2118 to GNP and AK), Rice University (Faculty Initiative award to GNP and AK), National Science Foundation (NSF), Directorate for Biological Sciences (grant No. 1231306 to GNP), an NSF CAREER award (No. 2145629 to AK), a Rice InterDisciplinary Excellence Award (IDEA), an Amazon Research Award and a Microsoft Research Award. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funders.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A., Bambrick, J., Bodenstein, S., Evans, D., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D. & Jumper, J. M. (2024). *Nature*, **630**, 493–500.

Agirre, J., Atanasova, M., Bagdonas, H., Ballard, C. B., Baslé, A., Beilsten-Edmands, J., Borges, R. J., Brown, D. G., Burgos-Mármol, J. J., Berrisford, J. M., Bond, P. S., Caballero, I., Catapano, L., Chojnowski, G., Cook, A. G., Cowtan, K. D., Croll, T. I., Debreczeni, J. É., Devenish, N. E., Dodson, E. J., Drevon, T. R., Emsley, P., Evans, G., Evans, P. R., Fando, M., Foadi, J., Fuentes-Montero, L., Garman, E. F., Gerstel, M., Gildea, R. J., Hatti, K., Hekkelman, M. L., Heuser, P., Hoh, S. W., Hough, M. A., Jenkins, H. T., Jiménez, E., Joosten, R. P., Keegan, R. M., Keep, N., Krissinel, E. B., Kolenko, P., Kovalevskiy, O., Lamzin, V. S., Lawson, D. M., Lebedev, A. A., Leslie, A. G. W., Lohkamp, B., Long, F., Malý, M., McCoy, A. J., McNicholas, S. J., Medina, A., Millán, C., Murray, J. W., Murshudov, G. N., Nicholls, R. A., Noble, M. E. M., Oeffner, R., Pannu, N. S., Parkhurst, J. M., Pearce, N., Pereira, J., Perrakis, A., Powell, H. R., Read, R. J., Rigden, D. J., Rochira, W., Sammito, M., Sánchez Rodríguez, F., Sheldrick, G. M., Shelley, K. L., Simkovic, F., Simpkin, A. J., Skubak, P., Sobolev, E., Steiner, R. A., Stevenson, K., Tews, I., Thomas, J. M. H., Thorn, A., Valls, J. T., Uski, V., Usón, I., Vagin, A., Velankar, S., Vollmar, M., Walden, H., Waterman, D., Wilson, K. S., Winn, M. D., Winter, G., Wojdyr, M. & Yamashita, K. (2023). *Acta Cryst.* D**79**, 449–461.

Armstrong, D. R., Berrisford, J. M., Conroy, M. J., Gutmanas, A., Anyango, S., Choudhary, P., Clark, A. R., Dana, J. M., Deshpande, M., Dunlop, R., Gane, P., Gáborová, R., Gupta, D., Haslam, P., Koča, J., Mak, L., Mir, S., Mukhopadhyay, A., Nadzirin, N., Nair, S., Paysan-Lafosse, T., Pravda, L., Sehnal, D., Salih, O., Smart, O., Tolchard, J., Varadi, M., Svobodova-Vařeková, R., Zaki, H., Kleywegt, G. J. & Velankar, S. (2019). *Nucleic Acids Res.* **48**, D335–D343.

Ashcroft, N. W. & Mermin, N. D. (2022). *Solid State Physics*. Boston: Cengage Learning.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán,

C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., Opperman, D. J., Sagmeister, T., Buhlheller, C., Pavkov-Keller, T., Rathinaswamy, M. K., Dalwadi, U., Yip, C. K., Burke, J. E., Garcia, K. C., Grishin, N. V., Adams, P. D., Read, R. J. & Baker, D. (2021). *Science*, **373**, 871–876.
Carrozzini, B., De Caro, L., Giannini, C., Altomare, A. & Caliandro, R. (2025). *Acta Cryst.* A**81**, 188–201.
Cowtan, K. (2011). *CPHASEMATCH*. https://www.ccp4.ac.uk/html/cphasematch.html.
Dana, J. M., Gutmanas, A., Tyagi, N., Qi, G., O'Donovan, C., Martin, M. & Velankar, S. (2019). *Nucleic Acids Res.* **47**, D482–D489.
Defazio, A., Yang, X., Mehta, H., Mishchenko, K., Khaled, A. & Cutkosky, A. (2024). *arXiv*:2405.15682.
Drenth, J. (2007). *Principles of Protein X-ray Crystallography*, 3rd ed. New York: Springer.
Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., Wiewiora, R. P., Brooks, B. R. & Pande, V. S. (2017). *PLoS Comput. Biol.* **13**, e1005659.
Jin, S., Miller, M. D., Chen, M., Schafer, N. P., Lin, X., Chen, X., Phillips, G. N. & Wolynes, P. G. (2020). *IUCrJ*, **7**, 1168–1178.
Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P. & Hassabis, D. (2021). *Nature*, **596**, 583–589.
Lattman, E. & Loll, P. (2008). *Protein Crystallography*. Baltimore: Johns Hopkins University Press.
Li, S., Zhao, Y., Varma, R., Salpekar, O., Noordhuis, P., Li, T., Paszke, A., Smith, J., Vaughan, B., Damania, P. & Chintala, S. (2020). *Proc. VLDB Endow.* **13**, 3005–3018.
Liebschner, D., Afonine, P. V., Baker, M. L., Bunkóczi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L.-W., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V., Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J. & Adams, P. D. (2019). *Acta Cryst.* D**75**, 861–877.
Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S. & Rives, A. (2023). *Science*, **379**, 1123–1130.
Matinyan, S., Filipcik, P. & Abrahams, J. P. (2024). *Acta Cryst.* A**80**, 1–17.
McCoy, A. J., Grosse-Kunstleve, R. W., Adams, P. D., Winn, M. D., Storoni, L. C. & Read, R. J. (2007). *J. Appl. Cryst.* **40**, 658–674.
McCoy, A. J., Sammito, M. D. & Read, R. J. (2022). *Acta Cryst.* D**78**, 1–13.
Needleman, S. B. & Wunsch, C. D. (1970). *J. Mol. Biol.* **48**, 443–453.
Pan, T., Dramko, E., Miller, M. D., Phillips, G. N. & Kyrillidis, A. (2025). *The Second Conference on Parsimony and Learning (Proceedings Track)*, https://openreview.net/forum?id=U9DhMKzXPT.
Pan, T., Dun, C., Jin, S., Miller, M. D., Kyrillidis, A. & Phillips, G. N. (2024). *Struct. Dyn.* **11**, 044701.
Pan, T., Jin, S., Miller, M. D., Kyrillidis, A. & Phillips, G. N. (2023). *IUCrJ*, **10**, 487–496.
Patterson, A. L. (1934). *Phys. Rev.* **46**, 372–376.
Smith, L. N. & Topin, N. (2019). *Proc. SPIE*, **11006**, 1100612.
Tanford, C. & Reynolds, J. (2004). *Nature's Robots: A History of Proteins*. Oxford University Press.
Terwilliger, T. C., Afonine, P. V., Liebschner, D., Croll, T. I., McCoy, A. J., Oeffner, R. D., Williams, C. J., Poon, B. K., Richardson, J. S., Read, R. J. & Adams, P. D. (2023). *Acta Cryst.* D**79**, 234–244.
Varadi, M., Bertoni, D., Magana, P., Paramval, U., Pidruchna, I., Radhakrishnan, M., Tsenkov, M., Nair, S., Mirdita, M., Yeo, J., Kovalevskiy, O., Tunyasuvunakool, K., Laydon, A., Žídek, A., Tomlinson, H., Hariharan, D., Abrahamson, J., Green, T., Jumper, J., Birney, E., Steinegger, M., Hassabis, D. & Velankar, S. (2024). *Nucleic Acids Res.* **52**, D368–D375.
Wimmer, T., Golkov, V., Dang, H. N., Zaiss, M., Maier, A. & Cremers, D. (2023). *arXiv*:2304.05864.
Wohlwend, J., Corso, G., Passaro, S., Getz, N., Reveiz, M., Leidal, K., Swiderski, W., Atkinson, L., Portnoi, T., Chinn, I., Silterra, J., Jaakkola, T. & Barzilay, R. (2025). *bioRxiv*, 2024.11.19.624167.
Wojdyr, M. (2022). *J. Open Source Softw.* **7**, 4200.
Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y. & Singh, V. (2021). *Proc. AAAI Conf. Artif. Intell.* **35**, 14138–14148.
Zhang, Y. & Yan, R. (2010). *NW-align*, Java version. https://zhanggroup.org/NW-align.