CrysFormer: Protein Crystallography Prediction via 3d Patterson Maps and Partial Structure Attention

Chen Dun
Rice University
cd46@rice.edu

Tom Pan
Rice University
qp3@rice.edu

Shikai Jin
Rice University
sj52@rice.edu

Ria Stevens
Rice University
rs127@rice.edu

Mitchell D. Miller
Rice University
mitchm@rice.edu

George N. Phillips, Jr.
Rice University
georgep@rice.edu

Anastasios Kyrillidis
Rice University
anastasios@rice.edu

Abstract

Determining the structure of a protein has been a decades-long open question. A protein's three-dimensional structure often poses nontrivial computation costs, when classical simulation algorithms are utilized. Advances in the transformer neural network architecture achieve significant improvements for this problem, by learning from a large dataset of sequence information and corresponding protein structures. Yet, such methods often only focus on sequence information; other available prior knowledge, such as protein crystallography and partial structure of amino acids, could be potentially utilized. To the best of our knowledge, we propose the first transformer-based model that directly utilizes protein crystallography and partial structure information to predict the electron density maps of proteins. Via two new datasets of peptide fragments (2-residue and 15-residue), we demonstrate our method, dubbed CrysFormer, can achieve accurate predictions, based on a much smaller dataset size and with reduced computation costs.

1 Introduction

Over the past decades, biologists have aimed to establish a standardized approach for experimentally determining and visualizing the overall structure of a protein at a low cost. There have been three general approaches to the protein structure problem: i) ones that rely on experimental measurements, such as X-ray crystallography, NMR, or cryo-electron microscopy; see [1]; ii) protein folding simulation tools based on thermodynamic or kinetic simulation of protein physics [2, 3]; and, iii) evolutionary programs based on bioinformatics analysis of the evolutionary history of proteins [4, 5].

Recent advances in machine learning (ML) algorithms have inspired a fourth direction which is to train a deep neural network model on a combination of a large-scale protein structure data set (i.e., the Protein Data Bank [6]) and knowledge of the amino acid sequences of a vast number of homologous proteins, to directly predict the protein structure from the protein's amino acid sequence. Recent research projects –such as Alphafold2 [7]– further show that, with co-evolutionary bioinformatic information, deep learning can achieve highly accurate predictions in most cases.

Our hypothesis and contributions. While it is true that such computational methods are improving, they are not yet complete –in terms of the types of structures that can be predicted– and suffer from a lack of accuracy in many details [8]. X-ray crystallographic data continues to be a gold standard for critical details describing chemical interactions of proteins. Having a robust and accurate way of going directly from an X-ray diffraction pattern to a solved structure would be a strong contribution to this field. Such approaches are missing from the literature, with the exception of [9].

Here, we present the first transformer-based model that utilizes protein crystallography and partial structure information to directly predict the electron density maps of proteins, going one step beyond

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

such recent approaches. While not yet ready to solve real problems, we demonstrate success on a simplified problem. As a highlight, using a new dataset of small peptide fragments of variable unit cell sizes –a byproduct of this work– we demonstrate that our method, named `CrysFormer`, can achieve more accurate predictions than recent work [9] with less computational overheads.

2 Problem Setup and Related Work

X-ray crystallography and the crystallographic phase problem. Each spot (known as a reflection) in an X-ray crystallography diffraction pattern is denoted by three indices h, k, l, known as Miller indices [10]. These correspond to sets of parallel planes within the protein crystal's unit cell that contribute to producing the reflections. The set of possible h, k, l values is determined by the radial extent of the observed diffraction pattern. Any reflection has an underlying mathematical representation, known as a structure factor, dependent on the locations and scattering factors of all the atoms within the crystal's unit cell. In math:

$$F(h,k,l) = \sum_{j=1}^{n} f_j \cdot e^{2\pi i(hx_j+ky_j+lz_j)}, \tag{1}$$

where the scattering factor and location of atom j are f_j and (x_j, y_j, z_j), respectively. A structure factor $F(h, k, l)$ has both an amplitude and a phase component (denoted by ϕ) and thus can be considered a complex number. Furthermore, suppose we knew both components of the structure factors corresponding to all of the reflections within a crystal's diffraction pattern. Then, in order to produce an accurate estimate of the electron density at any point (x, y, z) within the crystal's unit cell, we would only need to take a Fourier transform of all of these structures, as in:

$$\rho(x,y,z) = \tfrac{1}{V} \cdot \sum_{h,k,l} |F(h,k,l)| \cdot e^{-2\pi i(hx+ky+lz-\phi(h,k,l))}, \tag{2}$$

where V is the volume of the unit cell. The amplitude $|F(h, k, l)|$ of any structure factor is easy to determine, as it is simply proportional to the square root of the measured intensity of the corresponding reflection. However, it is impossible to directly determine the phase $\phi(h, k, l)$ of a structure factor, and this is what is well-known as the crystallographic phase problem [11].

Solving the phase problem. Various methods have been developed to solve the crystallography phase problem. The three commonly used methods are isomorphous replacement, anomalous scattering, and molecular replacement [11, 12]. Also, what is known as direct methods have been successful for small molecules that diffract to atomic resolution, but they rarely work for protein crystallography, due to the difficulty of resolving atoms as separate objects. Alternative methods have been developed to solve the phase problem based on intensity measurements alone, known as phase retrieval [13–15]. However, these methods have not been widely used in X-ray crystallography, because they assume different sampling conditions or were designed for non-crystallographic fields of physics. The iterative non-convex Gerchberg–Saxton algorithm [16, 17] is a well-known example of such methods, but requires more measurements than is available in crystallography.

3 `CrysFormer`: Using 3d Maps and Partial Structure Attention

The Patterson function. We utilize the *Patterson function* [18], a simplified variation of the Fourier transform from structure factors to electron density, in which all structure factor amplitudes are squared, and all phases are set to zero (i.e., ignored), as in:

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

It is important to note that a Patterson map can be directly obtained from raw diffraction data without the need for additional experiments, or any other information. And due to the discrete size of the input and output layers in deep learning models, we can discretize and reformulate the electron density map –and its corresponding Patterson map– as follows: Suppose the electron density map of a molecule in interest is discretized into a $N_1 \times N_2 \times N_3$ 3d grid. The electron density map can then be denoted as $\mathbf{e} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$. The Patterson map is then formulated as follows, where $\odot$ means matrix element-wise multiplication:

$$\mathbf{p} = \Re\left(\mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{e}) \odot \mathcal{F}(\widehat{\mathbf{e}})\right)\right) \approx \Re\left(\mathcal{F}^{-1}\left(|\mathcal{F}(\mathbf{e})|^2\right)\right).$$

Breaking down the above expression, $\mathcal{F}(\mathbf{e}) \odot \mathcal{F}(\widehat{\mathbf{e}}) \approx |\mathcal{F}(\mathbf{e})|^2$ denotes only the magnitude part of the complex signals, as measured through the Fourier transform of the input signal $\mathbf{e}$. Here, $\widehat{\mathbf{e}}$ denotes an inverse-shifted version of $\mathbf{e}$, where its entries follow the shifted rule as in $\widehat{e}_{i,j,k} = e_{N-i,N-j,N-k}$.

Using deep learning. We follow a data-centric approach and train a deep learning model, abstractly represented by $g(\boldsymbol{\theta}, \cdot)$, such that given a Patterson map $\mathbf{p}$ as input, it generates an estimate of an electron density map, that resembles closely the true map $\mathbf{e}$. Formally, given a data distribution $\mathcal{D}$ and $\{\mathbf{p}_i, \mathbf{e}_i\}_{i=1}^n \sim \mathcal{D}$, where $\mathbf{p}_i \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is the Patterson map that corresponds to the true data electron density map, $\mathbf{e}_i \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, deep learning training aims in finding $\boldsymbol{\theta}^\star$ as in:

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \left\{ \mathcal{L}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{\theta};\ g, \{\mathbf{p}_i, \mathbf{e}_i\}) = \frac{1}{n} \sum_{i=1}^{n} \|g(\boldsymbol{\theta}, \mathbf{p}_i) - \mathbf{e}_i\|_2^2 \right\}.$$

Since we have a regression problem, we use mean squared error as the loss function $\mathcal{L}(\boldsymbol{\theta})$.

Using partial protein structures. Due to the well-studied structure of amino acids, we aim to optionally utilize standardized *partial structures* to aid prediction, when they are available. For example, let $\mathbf{u}_i^j \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ be the known standalone electron density map of the j-th amino acid of the i-th protein sample, in a standardized conformation. Abstractly, we then aim to optimize:

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \left\{ \mathcal{L}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{\theta};\ g, \{\mathbf{p}_i, \mathbf{e}_i, \mathbf{u}_i^j\}) = \frac{1}{n} \sum_{i=1}^{n} \|g(\boldsymbol{\theta}, \mathbf{p}_i, \mathbf{u}_i^j) - \mathbf{e}_i\|_2^2 \right\}.$$

Our proposal. We propose `CrysFormer`, a novel, 3d Transformer model [19, 20] with a new self-attention mechanism to process Patterson maps and partial protein structures, to directly infer electron density maps with reduced costs. `CrysFormer` captures the global information in Patterson maps and "translates" it into correct electron density map predictions, via our proposed self-attention mechanism. `CrysFormer` does not need an encoder-decoder structure [19] and artificial information bottlenecks [21] –as in the `U-Net` architecture– to force the learning of global information. `CrysFormer` is able to handle additional partial structure information, which comes from a different domain than the Patterson maps. By using efficient self-attention between 3d image patches, we can significantly reduce our overall computation costs.

The architecture of the `CrysFormer`. We follow ideas of a 3d visual Transformer [20] by partitioning the whole input 3d Patterson map $\mathbf{p}_i \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ input into a set of smaller 3d patches. We embed them into one-dimensional "word tokens", and feed them into a multi-layer, encoder-only Transformer module. If partial structures $\mathbf{u}_i^j$ are also available, we will partition them into 3d patches and embed them into additional tokens that are sent to each self-attention layer. This way, the tokens in each layer can also "attend" the election density of partial structures, as a reference for final global electron density map predictions. Finally, we utilize a 3d convolutional layer to transform "word-tokens" back into a 3d electron density map.[1]; see Figure 1. A precise mathematical formulation of our model architecture is found in the appendix.

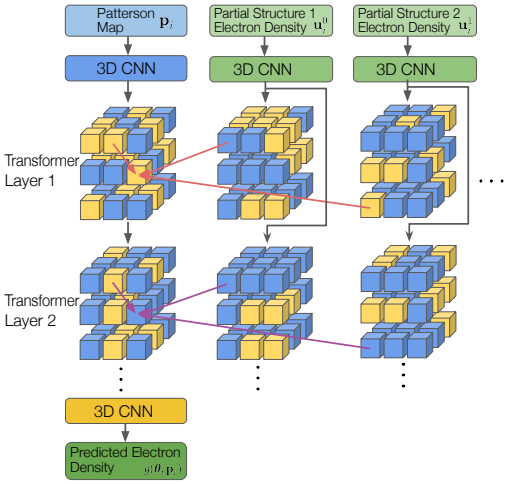


Figure 1: `Crysformer` and its novel one-way attention mechanism for partial structures (red and purple arrows).

4 New Datasets

We generate datasets of protein fragments, where input Patterson and output electron density maps are derived from Protein Databank (PDB) entries of proteins solved by X-ray Crystallography [6]. We start from a curated basis of $\sim 24,000$ such protein structures. Then from a random subset of about half of these structures, we randomly select and store segments of adjacent amino acid residues. These examples are consisted of dipeptides (two residues) and 15-residues, leading to two datasets that we introduce with this work. The latter dataset contains 15 residues, where at most 3 residues could be shared between different examples. Using the `pdbfixer` Python API [22], we remove all examples that either contain nonstandard residues.

[1]We also utilize 3d convolutional layer(s) at the very beginning of the execution to expand the number of channels of the Patterson map (and potentially partial structure) inputs.

For our dipeptide dataset, we then iteratively expand the unit cell dimensions for each example, starting from the raw $\max - \min$ ranges in each of the three axis directions, attempting to create a minimal-size unit cell where the minimum atomic contact is at least 2.75 Angstroms (Å). For our 15-residue dataset, we instead place atoms in fixed unit cells of size 41 Å x 30 Å x 24 Å to simplify the now much harder problem. After this, all examples that still contain atomic contacts of less than 2.75 Å are discarded. The examples are then reoriented via a reindexing operation, such that the first axis is always the longest and the third axis is always the shortest. We also center all atomic coordinates such that the center of mass is in the center of the corresponding unit cell.

Structure factors for each remaining example, as well as those for the corresponding partial structures for each of the present amino acids, are generated using the `gemmi sfcalc` program [23] to a resolution of 1.5 Å. An electron density and Patterson map for each example are then obtained from those structure factors with the `fft` program of the `CCP4` program suite [24, 25]; partial structure densities are obtained in the same manner. We specify a grid oversampling factor of 3.0, resulting in a 0.5 Å grid spacing in the produced maps. All these maps are then converted into PyTorch tensors.

5 Experiments

Baselines. There are no readily available off-the-self solutions for our setting, as our work is one of the first of this kind. As our baseline, we use a CNN-based `U-Net` model [9]; this architecture is widely used in image transformation tasks [26, 27].

For comparison, we have further enhanced this vanilla `U-Net` with $i)$ additional input channels to incorporate the partial structure information, despite being evidently unsound; and $ii)$ a refining model procedure, which retrains the `U-Net` using previous model predictions as additional input channels. Both of these extensions are shown to greatly improve the performance of the vanilla `U-Net`. We refer the reader to the appendix for more details on our baseline model architecture.

Metrics. During testing, we calculate the Pearson correlation coefficient between the ground truth targets $\mathbf{e}$ and model predictions $g(\boldsymbol{\theta}, \mathbf{p})$; the larger this coefficient is, the better. Let us denote a model prediction as $\mathbf{e}'$. We define $\bar{\mathbf{e}} = \frac{1}{N_1 N_2 N_3} \sum_{i,j,k} \mathbf{e}_{i,j,k}$ and $\bar{\mathbf{e}}' = \frac{1}{N_1 N_2 N_3} \sum_{i,j,k} \mathbf{e}'_{i,j,k}$. Then, the Pearson correlation coefficient between $\mathbf{e}$ and $\mathbf{e}'$ is as below:

$$\texttt{PC}(\mathbf{e}, \mathbf{e}') = \frac{\sum_{i,j,k=1}^{N_1,N_2,N_3} (\mathbf{e}'_{i,j,k} - \bar{\mathbf{e}}')(\mathbf{e}_{i,j,k} - \bar{\mathbf{e}})}{\sqrt{\sum_{i,j,k=1}^{N_1,N_2,N_3} (\mathbf{e}'_{i,j,k} - \bar{\mathbf{e}}') + \epsilon} \cdot \sqrt{\sum_{i,j,k=1}^{N_1,N_2,N_3} (\mathbf{e}_{i,j,k} - \bar{\mathbf{e}}) + \epsilon}}, \tag{4}$$

where ϵ is a small constant to prevent division by zero. To demonstrate how well our methods solve the phase problem, we also perform phase error analysis on our models' final post-training predictions using the `cphasematch` program of the `CCP4` program suite [28]. We report the mean phase errors of our predictions in degrees, as reported by `cphasematch`, where a smaller phase error is desirable. Finally, we compare the convergence speed and computation cost of both methods.

Method	Mean $\texttt{PC}(\mathbf{e}, \mathbf{e}')$	Mean Phase Error	Epochs	Time per epoch (mins.)
`U-Net` [9]	0.735	67.40°	50	28.93
`U-Net+R` (This work)	0.775	58.67°	90	29.06
`U-Net+PS+R` (This work)	0.839	51.34°	90	29.31
`CrysFormer` (This work)	**0.939**	**35.16°**	**35**	**12.37**

Table 1: `CrysFormer` versus baselines on the dipeptide dataset. `U-Net+R` refers to adding the refining procedure to `U-Net` training; `U-Net+PS+R` refers to adding further partial structures as additional channels.

Results on two-residues. A summary of our results on our dipeptide dataset, which consisted of $1,894,984$ training and $210,487$ test cases, is provided in Table 1. Overall, `CrysFormer` achieves a significant improvement in prediction accuracy in terms of both the Pearson coefficient and phase error, while requiring a shorter time (in epochs) to converge. `CrysFormer` also incurs much less computation cost which results in significantly reduced wall clock time per epoch.

We further plot the calculated average mean phase errors of the predictions of our models against reflection resolution, see left panel of Figure 2. The predictions made by `CrysFormer` have lower mean phase error, compared to baselines. This means that the `CrysFormer` predictions, on average, can reproduce better the general shape, as well as finer details of the ground truth electron densities.

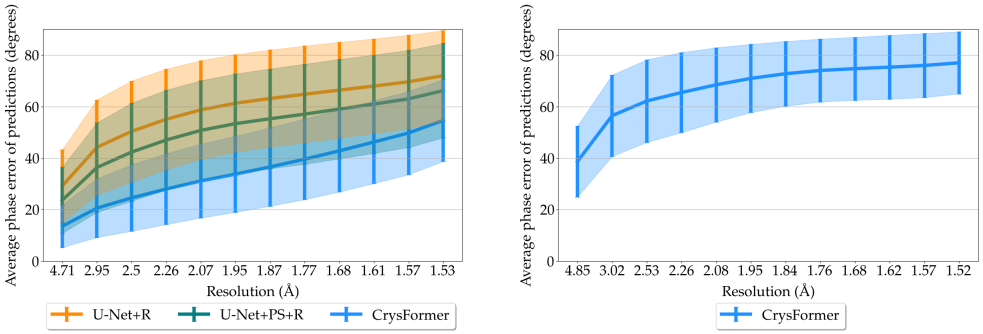
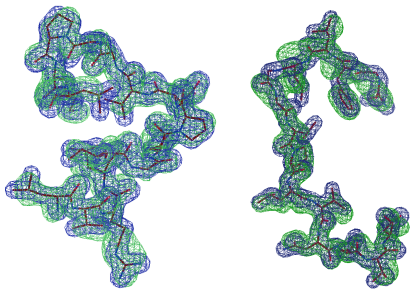


Figure 2: **Left**: Average phase error of model predictions against reflection resolution for dipeptide dataset. **Right**: Average phase error of model predictions against reflection resolution for 15-residue dataset.

Results on 15-residues. On our dataset of 15-residue examples, which consisted of only $165,858$ training and $16,230$ test cases, we trained for 80 epochs to a final average test set Pearson correlation of about 0.747. We then performed a refining training run of 20 epochs, incorporating the original training run's predictions as additional input channels when training the `CrysFormer`, and obtained an improved average test set Pearson correlation of about 0.77 and phase error of about 67.66. On both of these runs, we used the Nyström approximate attention mechanism [29] when incorporating our partial structure information to reduce time and space costs. Even still, each training epoch still took about 6.28 hours to complete. Thus due to time considerations, we decided not to attempt to train a U-Net on this dataset for purposes of comparison.

We provide visualizations of a model prediction in Figure 3; more can again be found in the appendix. We also plot the average mean phase errors of the predictions of our models against reflection resolution, shown on the right in Figure 2. These results show that this is a more difficult dataset with reduced sample size; yet `CrysFormer` predictions tend to reproduce details of the desired electron densities.



(a) `5E4V_1_25` (b) `3HN7_1_133`

Figure 3: Visualization of two successful predictions after a refining training run; ground truth density maps shown in blue and predictions shown in green.

We used the *Autobuild* program within the *PHENIX* suite [30, 31] to perform automated model building and crystallographic refinement on a randomly selected subset of 302 test set predictions after the refining training run. We found that 281 out of 302 ($\sim 93\%$) refined to a final atomic model with a crystallographic R-factor of less than 0.38, indicating success, when solvent flattening was applied. Without solvent flattening, 258 out of 302 ($\sim 85\%$) refined to such an R-factor (performing solvent flattening is known to be especially effective for unit cells with high solvent content). And even if no refinement was performed at all, and instead an atomic model was repeatedly fit to our predicted electron densities, we found that 229 out of 302 ($\sim 76\%$) of the best such atomic models still had a crystallographic R-factor of less than 0.38.

Furthermore, after automatic map interpretation using the autobuilding routines in *shelxe* [32] to obtain a poly-alanine chain from each of the 16230 test set predictions, we found that almost 74% of the resulting models had calculated amplitudes with a Pearson correlation of at least 0.25 to the true underlying data. Historical results indicate that further refinement would very likely produce a "correct" model if the initial poly-alanine model has at least such a correlation.

6 Conclusion

We have shown that `CrysFormer` outperforms recent models for predicting electron density maps from corresponding Patterson maps in all metrics on a newly introduced dataset (dipeptide). Also, `CrysFormer` requires fewer epochs to reasonably converge and has a smaller computational footprint. Furthermore, our "refining" procedure greatly improves training for the vanilla `U-Net` architecture on our dipeptide dataset, as well as for training `CrysFormer` on our 15-residues dataset.

Acknowledgements

This research was funded in part by: The Robert A. Welch Foundation (grant No. C-2118 to G.N.P and A.K.); NSF, Directorate for Biological Sciences (grant No. 1231306 to G.N.P.); Rice University (Faculty Initiative award to G.N.P and A.K.); NSF FET:Small (award no. 1907936); NSF MLWiNS CNS (award no. 2003137, in collaboration with Intel); NSF CMMI (award no. 2037545); NSF CAREER (award no. 2145629); a Rice InterDisciplinary Excellence Award (IDEA); an Amazon Research Award; 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

[1] Jan Drenth. *Principles of protein X-ray crystallography*. Springer Science & Business Media, 2007. doi:10.1007/0-387-33746-6.

[2] Emiliano Brini, Carlos Simmerling, and Ken Dill. Protein storytelling through physics. *Science*, 370(6520):eaaz3041, 2020. doi:10.1126/science.aaz3041.

[3] Manfred J Sippl. Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *Journal of molecular biology*, 213(4):859–883, 1990. doi:10.1016/s0022-2836(05)80269-4.

[4] Andrej Šali and Tom L Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of molecular biology*, 234(3):779–815, 1993. doi:10.1006/jmbi.1993.1626.

[5] Ambrish Roy, Alper Kucukural, and Yang Zhang. *I-TASSER*: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4):725–738, 2010. doi:10.1038/nprot.2010.5.

[6] wwPDB consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Research*, 47(D1):D520–D528, 2019. doi:10.1093/nar/gky949.

[7] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. doi:10.1038/s41586-021-03819-2.

[8] Thomas C. Terwilliger, Dorothee Liebschner, Tristan I. Croll, Christopher J. Williams, Airlie J. McCoy, Billy K. Poon, Pavel V. Afonine, Robert D. Oeffner, Jane S. Richardson, Randy J. Read, and Paul D. Adams. Alphafold predictions are valuable hypotheses and accelerate but do not replace experimental structure determination. *Nature Methods*, 2023. doi:10.1038/s41592-023-02087-4.

[9] Tom Pan, Shikai Jin, Mitchell D Miller, Anastasios Kyrillidis, and George N Phillips. A deep learning solution for crystallographic structure determination. *IUCrJ*, 10(4):487–496, 2023. doi:10.1107/S2052252523004293.

[10] Neil W Ashcroft and N David Mermin. *Solid state physics*. Cengage Learning, 2022. ISBN 0357886089.

[11] Eaton Lattman and Patrick Loll. *Protein Crystallography*. Johns Hopkins University Press, 2008. ISBN 0801888085.

[12] Shikai Jin, Mitchell D Miller, Mingchen Chen, Nicholas P Schafer, Xingcheng Lin, Xun Chen, GN Phillips, and PG Wolynes. Molecular-replacement phasing using predicted protein structures from awsem-suite. *IUCrJ*, 7(6):1168–1178, 2020. doi:10.1107/s2052252520013494.

[13] Youming Guo, Yu Wu, Ying Li, Xuejun Rao, and Changhui Rao. Deep phase retrieval for astronomical Shack–Hartmann wavefront sensors. *Monthly Notices of the Royal Astronomical Society*, 510(3):4347–4354, 12 2021. ISSN 0035-8711. doi:10.1093/mnras/stab3690.

[14] Armin Kappeler, Sushobhan Ghosh, Jason Holloway, Oliver Cossairt, and Aggelos Katsaggelos. Ptychnet: CNN based fourier ptychography. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1712–1716, New York, NY, USA, 2017. IEEE Press. doi:10.1109/ICIP.2017.8296574.

[15] Yair Rivenson, Yibo Zhang, Harun Günaydın, Da Teng, and Aydogan Ozcan. Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light: Science & Applications*, 7(2):17141–17141, 2018. doi:10.1038/lsa.2017.141.

[16] J. R. Fienup. Phase retrieval algorithms: a comparison. *Appl. Opt.*, 21(15):2758–2769, Aug 1982. doi:10.1364/AO.21.002758.

[17] Zeev Zalevsky, David Mendlovic, and Rainer G Dorsch. Gerchberg–saxton algorithm applied in the fractional fourier or the fresnel domain. *Optics Letters*, 21(12):842–844, 1996. doi:10.1364/ol.21.000842.

[18] A. L. Patterson. A fourier series method for the determination of the components of interatomic distances in crystals. *Phys. Rev.*, 46:372–376, Sep 1934. doi:10.1103/PhysRev.46.372.

[19] 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. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[20] Junyu Chen, Yufan He, Eric C. Frey, Ye Li, and Yong Du. ViT-V-Net: Vision transformer for unsupervised volumetric medical image registration. *arXiv*, .2104.06468, 2021. doi:10.48550/arXiv.2104.06468.

[21] Hao Cheng, Dongze Lian, Shenghua Gao, and Yanlin Geng. Utilizing information bottleneck to evaluate the capability of deep neural networks for image classification. *Entropy*, 21(5), 2019. ISSN 1099-4300. doi:10.3390/e21050456.

[22] Peter Eastman, Jason Swails, John D Chodera, Robert T McGibbon, Yutong Zhao, Kyle A Beauchamp, Lee-Ping Wang, Andrew C Simmonett, Matthew P Harrigan, Chaya D Stern, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7):e1005659, 2017. doi:10.1371/journal.pcbi.1005659.

[23] Marcin Wojdyr. Gemmi: A library for structural biology. *Journal of Open Source Software*, 7 (73):4200, 2022. doi:10.21105/joss.04200.

[24] RJ Read and AJ Schierbeek. A phased translation function. *Journal of Applied Crystallography*, 21(5):490–495, 1988. doi:10.1107/S002188988800562X.

[25] Martyn D. Winn, Charles C. Ballard, Kevin D. Cowtan, Eleanor J. Dodson, Paul Emsley, Phil R. Evans, Ronan M. Keegan, Eugene B. Krissinel, Andrew G. W. Leslie, Airlie McCoy, Stuart J. McNicholas, Garib N. Murshudov, Navraj S. Pannu, Elizabeth A. Potterton, Harold R. Powell, Randy J. Read, Alexei Vagin, and Keith S. Wilson. Overview of the *CCP*4 suite and current developments. *Acta Crystallographica Section D*, 67(4):235–242, Apr 2011. doi:10.1107/S0907444910045749.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241, 2015. doi:10.1007/978-3-319-24574-4_28.

[27] Weidan Yan, Can Chen, and Dengyin Zhang. U-Net-based medical image segmentation algorithm. In *13th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5, 2021. doi:10.1109/WCSP52459.2021.9613447.

[28] Kevin Cowtan. cphasematch, 2011. URL `https://www.ccp4.ac.uk/html/cphasematch.html`.

[29] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14138–14148, May 2021. doi:10.1609/aaai.v35i16.17664.

[30] Thomas C. Terwilliger, Ralf W. Grosse-Kunstleve, Pavel V. Afonine, Nigel W. Moriarty, Peter H. Zwart, Li-Wei Hung, Randy J. Read, and Paul D. Adams. Iterative model building, structure refinement and density modification with the *PHENIX AutoBuild* wizard. *Acta Crystallogr.*, D64(1):61–69, Jan 2008. doi:10.1107/S090744490705024X.

[31] Dorothee Liebschner, Pavel V. Afonine, Matthew L. Baker, Gábor Bunkóczi, Vincent B. Chen, Tristan I. Croll, Bradley Hintze, Li-Wei Hung, Swati Jain, Airlie J. McCoy, Nigel W. Moriarty, Robert D. Oeffner, Billy K. Poon, Michael G. Prisant, Randy J. Read, Jane S. Richardson, David C. Richardson, Massimo D. Sammito, Oleg V. Sobolev, Duncan H. Stockwell, Thomas C. Terwilliger, Alexandre G. Urzhumtsev, Lizbeth L. Videau, Christopher J. Williams, and Paul D. Adams. Macromolecular structure determination using x-rays, neutrons and electrons: recent developments in *phenix*. *Acta Crystallogr.*, D75(10):861–877, Oct 2019. doi:10.1107/S2059798319011471.

[32] Isabel Usón and George M. Sheldrick. An introduction to experimental phasing of macromolecules illustrated by *SHELX*; new autotracing features. *Acta Crystallogr.*, D74(2):106–116, Feb 2018. doi:10.1107/S2059798317015121.

[33] David Hurwitz. From patterson maps to atomic coordinates: Training a deep neural network to solve the phase problem for a simplified case. *arXiv*, 03 2020. doi:10.48550/arXiv.2003.13767.

[34] Anne Marie Helmenstine. Amino acid chirality, 2021. URL `https://www.thoughtco.com/amino-acid-chirality-4009939`.

APPENDIX

A Model Architecture of `CrysFormer`

The first part of our model is the preprocessing and partitioning of input Patterson maps $\mathbf{p}$ and additional partial structures $\mathbf{u}^j$ into 3d patches of size $d_1 \times d_2 \times d_3$. We embed those patches into one-dimensional tokens with dimension d_t, using of a small MLP, and add them with a learned positional embedding; this holds for both Patterson maps and structures, as below:

Patterson maps $\mathbf{p}$	**Partial structures** $\mathbf{u}^j$
$\mathbf{X}^0 = \texttt{3DCNN}_{\mathbf{W}_c}(\mathbf{p}) \in \mathbb{R}^{c\times N_1\times N_2\times N_3}$	$\mathbf{U}^j = \texttt{3DCNN}_{\mathbf{W}_p}(\mathbf{u}^j) \in \mathbb{R}^{c\times N_1\times N_2\times N_3}$
$\mathbf{X}^0 = \texttt{Partition}(\mathbf{X}^0) \in \mathbb{R}^{\frac{N_1}{d_1}\times\frac{N_2}{d_2}\times\frac{N_3}{d_3}\times(cd_1d_2d_3)}$	$\mathbf{U}^j = \texttt{Partition}(\mathbf{U}^j) \in \mathbb{R}^{\frac{N_1}{d_1}\times\frac{N_2}{d_2}\times\frac{N_3}{d_3}\times(cd_1d_2d_3)}$
$\mathbf{X}^0 = \texttt{Flatten}(\mathbf{X}^0) \in \mathbb{R}^{\frac{N_1N_2N_3}{d_1d_2d_3}\times(cd_1d_2d_3)}$	$\mathbf{U}^j = \texttt{Flatten}(\mathbf{U}^j) \in \mathbb{R}^{\frac{N_1N_2N_3}{d_1d_2d_3}\times(cd_1d_2d_3)}$
$\mathbf{X}^0 = \texttt{MLP}_{\mathbf{W}_c}(\mathbf{X}^0) \in \mathbb{R}^{\frac{N_1N_2N_3}{d_1d_2d_3}\times d_t}$	$\mathbf{U}^j = \texttt{MLP}_{\mathbf{W}_p}(\mathbf{U}^j) \in \mathbb{R}^{\frac{N_1N_2N_3}{d_1d_2d_3}\times d_t}$
$\mathbf{X}^0 = \mathbf{X}^0 + \texttt{PosEmbedding}(\frac{N_1N_2N_3}{d_1d_2d_3})$	$\mathbf{U}^j = \mathbf{U}^j + \texttt{PosEmbedding}(\frac{N_1N_2N_3}{d_1d_2d_3})$

As shown in Figure 1, we design an efficient attention mechanism such that i) only tokens from Patterson maps attend tokens from the partial structures; ii) the tokens from the additional partial structures are not passed to the next layer. This is based on that the partial structure electron density information should be used by the model as a stable reference to attend to in each layer.

This one-way attention also greatly reduces the overall communication cost. In particular, let the token sequence length be $S = \frac{N_1N_2N_3}{d_1d_2d_3}$ and let d_h denote the dimension of the attention head. Assuming we have H attention heads and L layers, `CrysFormer` uses the following attention mechanism:

$$
\begin{aligned}
\mathbf{U} &= \texttt{Concat}_{j=1}^{J}(\mathbf{U}^j) \in \mathbb{R}^{(SJ)\times d_t} \\
\mathbf{A}^h &= \texttt{Softmax}\left((\mathbf{W}_q^h\mathbf{X}^\ell)^\top(\texttt{Concat}(\mathbf{W}_k^h\mathbf{X}^\ell, \mathbf{W}_{k'}^h\mathbf{U})\right) \in \mathbb{R}^{S\times(S+SJ)}; \\
\widehat{\mathbf{V}}^h &= \mathbf{A}^h\left(\texttt{Concat}(\mathbf{W}_v^h\mathbf{X}^\ell, \mathbf{W}_{v'}^h\mathbf{U})\right) \in \mathbb{R}^{S\times d_h}; \\
\mathbf{O} &= \mathbf{W}_o\texttt{Concat}\left(\widehat{\mathbf{V}}^0,\ \widehat{\mathbf{V}}^1, \ldots,\ \widehat{\mathbf{V}}^H\right) \in \mathbb{R}^{S\times d_t}; \\
\mathbf{X}^{\ell+1} &= \mathbf{W}_{\text{ff2}}(\texttt{ReLU}(\mathbf{W}_{\text{ff1}}\mathbf{O})),
\end{aligned}
$$

where, omitting the layer index, $\mathbf{W}_q^h$, $\mathbf{W}_k^h$, $\mathbf{W}_v^h$ are the trainable query, key, and value projection matrices of the h-th attention head for tokens from the Patterson map, and $\mathbf{W}_{k'}^h$, $\mathbf{W}_{v'}^h$ are the corresponding matrices for tokens from the partial structure, each with dimension d_h. Further, $\mathbf{W}_{\text{ff1}}$ and $\mathbf{W}_{\text{ff2}}$ are the trainable parameters of the fully-connected layers. We omit skip connections and layer normalization modules just to simplify notation, but these are included in practice.

As a final step, we transform the output embedding back to a 3d electron density map, as follows:

$$
g(\boldsymbol{\theta}, \mathbf{p}) = \texttt{tanh}(\texttt{3DCNN}_{\mathbf{W}_o}(\texttt{Rearrange}(\texttt{MLP}(\mathbf{X}^L)))) \in \mathbb{R}^{N_1\times N_2\times N_3},
$$

As stated, we use as our loss function the standard mean squared error loss.

B Additional Details on Dataset Generation

To start preparing our dataset, we selected nearly 24000 representative Protein Data Bank (PDB) entries using the following criteria: proteins solved by X-ray crystallography after 1995, sequence length ≥ 40, refinement resolution ≤ 2.75, refinement R-Free ≤ 0.28, with clustering at 30% sequence identity. The standardized modifications we applied to each viable coordinate file were as follows: all temperature factors were set to 20, any selenomethionine residues were rebuilt as methionine, and all hydrogen atoms were removed leaving only carbon, nitrogen, oxygen, and potentially sulfur.

In our dataset generation process, an effort was taken to ensure diversity by sampling from PDB entities with low sequence similarity to each other. However, both test and training sets are taking random samples from the conformations allowed in rotamer and Ramachandran space. Any similar conformations would be expected to be in a different rotational orientation in the cell by the nature of the selection process. We did not compute all-versus-all clustering or force the test and training sets to sample distinct conformational regions. For our 15-residue dataset, in order to obtain a greater amount of starting coordinate files, we allowed at most 3 residues to be shared between distinct examples. To prevent potential overfitting that could arise from this sharing of subsegments, we enforced that all examples derived from the same initial .pdb file would be placed together in either the training or test set.

Another issue regarding ambiguity in Patterson map interpretation is the fact that an electron density will always have the exact same Patterson map as its corresponding centrosymmetry-related electron density. [33] provided a workaround that involved combining a set of atoms with the set of its centrosymmetry-related atoms into a single example output. However, this also requires a separate post-processing algorithm to separate the original and centrosymmetric densities for each of his model's predictions. Since we are working with real-world structures –rather than randomly placed data– we can exploit their known properties. In particular, we know that all proteinogenic amino acids are naturally found in only one possible enantiomeric configuration [34]. Although the mirror-image symmetry of enantiomers is not exactly the same as centrosymmetry, we show that this is enough to allow us to work with true electron densities of protein fragments.

C Description of Dataset Subset

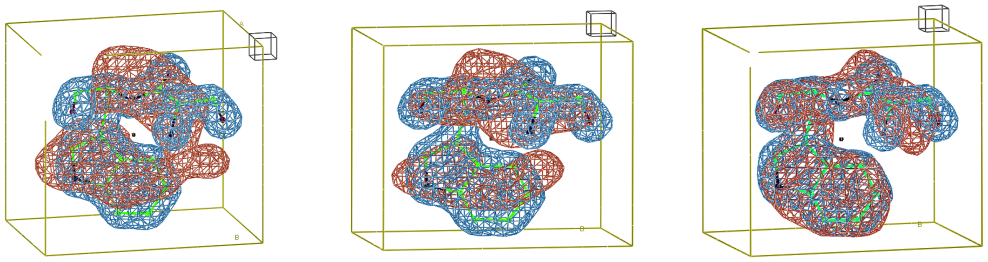
Due to limitations of online storage space, we provide a subset of our generated dataset. This subset represents a total of 200000 dipeptide examples. As expected, patterson.tar.gz contains the generated Patterson maps, while electron_density.tar.gz contains the corresponding electron densities. Meanwhile, partial_structure.tar.gz contains both of the partial structures for each dipeptide example in the subset.

The dataset can be downloaded through this link:

https://drive.google.com/drive/folders/1X7YkxDd7yTC1RTG1z3NbdRIfKLfFtkrx?usp=share_link

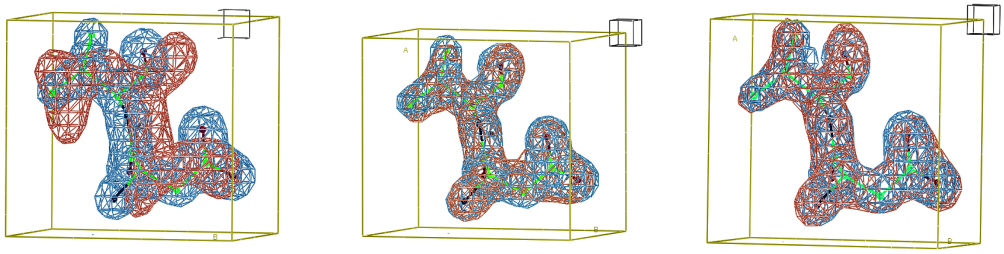
We will also provide a dataset of prepared .pdb coordinate files of 15-residue examples, to which our dataset generation process can be applied in order to produce Patterson map and electron density tensors.

D Additional Visualizations of Model Predictions



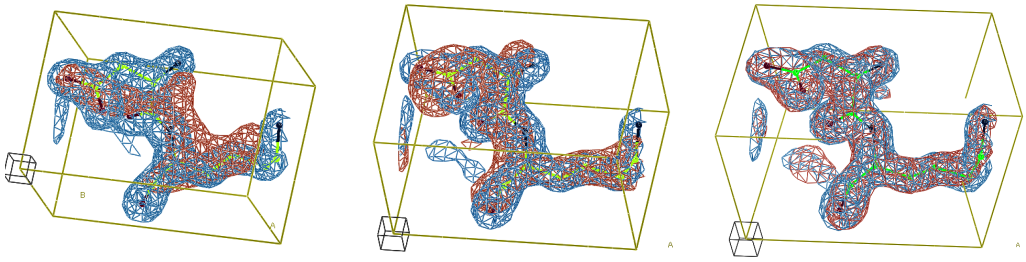
(a) U-Net+R (b) U-Net+PS+R (c) CrysFormer

Figure 4: Serine + Tryptophan



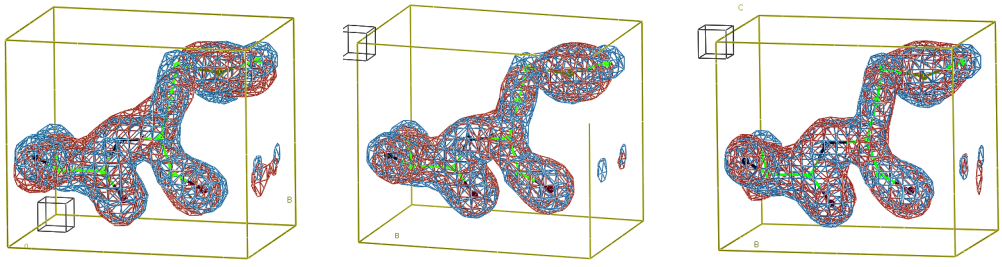
(a) U-Net+R (b) U-Net+PS+R (c) CrysFormer

Figure 5: Aspartic Acid + Valine



(a) U-Net+R (b) U-Net+PS+R (c) CrysFormer

Figure 6: Aspartic Acid + Lysine



(a) U-Net+R (b) U-Net+PS+R (c) CrysFormer

Figure 7: Alanine + Methionine

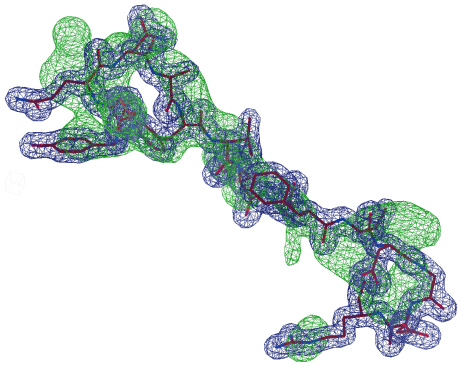
Figure 8: Visualizations for dipeptide dataset. Ground truth density maps are shown in blue, while predictions are shown in red. The model used to generate the ground truth electron density is shown in stick representation for reference.

Figure 4 represents a class of examples containing a large aromatic residue, Tryptophan. U-Net+R models consistently produce poor predictions in this case, while the CrysFormer better handles such residues. U-Net+PS+R shows that both providing additional input channels and using the refining procedure improves results even for U-Net architectures; yet, CrysFormer still provides better reconstruction. Figure 5 represents an example in which the additional partial structure input channels provided to the U-Net provided a substantial increase in prediction quality, allowing it to produce a prediction similar to that of the CrysFormer. Figure 6 represents an example in which both providing additional input channels to the U-Net and switching to CrysFormer provided noticeable improvements in prediction quality. Finally, Figure 7 represents an example in which all of our models provided a reasonably accurate prediction.

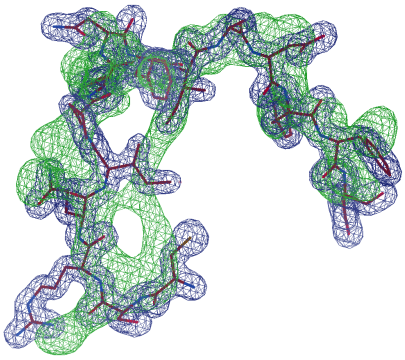
Figure 9 shows various CrysFormer on our 15-residue dataset. It is clear that as prediction quality increases as indicated by reported Pearson correlation, finer details of the true underlying structure are more likely to be accurately reproduced. The predictions in Figure 9 (e), (f), (g), and (h), as well as Figure 3 (a) [rank 55%] and (b) [rank 82%], were all successfully refined using all of the mentioned autotracing and refinement procedures. But even for relatively poor predictions such as (a) and (b), the rough overall shape can be reproduced even though several portions have clear inaccuracies.

Figure 10 shows the results of our *Autobuild* refinement runs as scatterplots; clearly only a small fraction of the subset of predictions did not refine successfully.

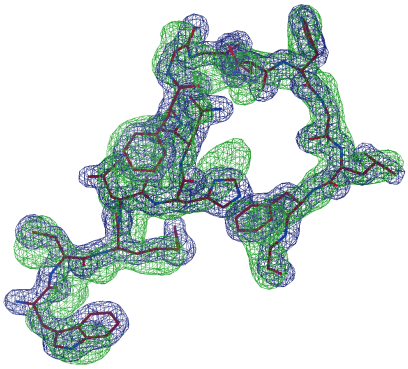
Figure 11 shows the scatterplot of *shelxe* poly-alanine autotracing results on the full 15-residue test set. As mentioned, examples for which the amplitudes calculated from the initial poly-alanine chain built into the model electron density prediction have a Pearson correlation coefficient with the true underlying structure factor amplitudes of over 0.25 (shown above the red line) are extremely likely to be successfully refined.



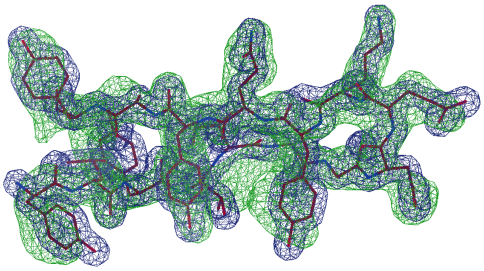
(a) 4KNK_1.pd_73 CC 0.60 (Rank 11%)



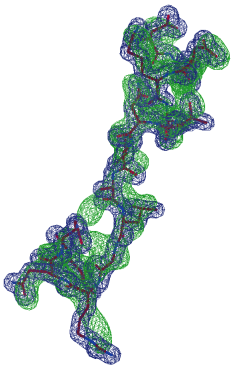
(b) 7F1T_1.pd_13 CC 0.66 (Rank 18%)



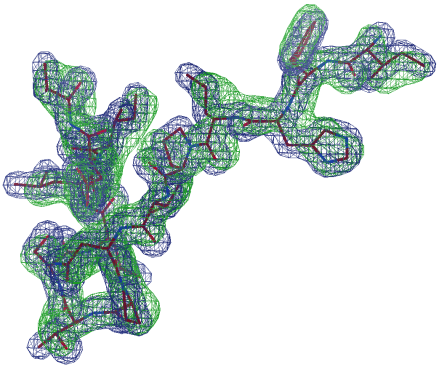
(c) 4XWH_1.pd_380 CC 0.75 (Rank 31%)



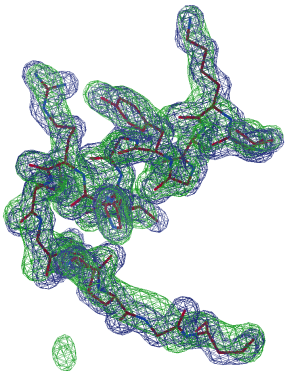
(d) 5MSX_1.pd_193 CC 0.76 (Rank 36%)



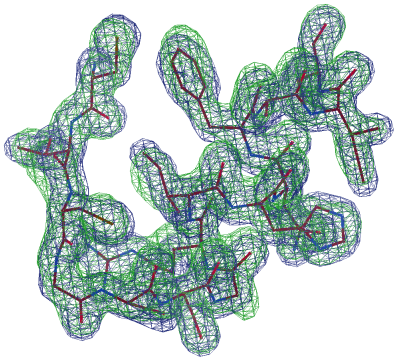
(e) 4FBC_1.pd_121 CC 0.78 (Rank 38%)



(f) 7K34_1.pd_145 CC 0.84 (Rank 57%)



(g) 7F1T_1.pd_13 CC 0.87 (Rank 63%)



(h) 4TXJ_1.pd_37 CC 0.92 (Rank 90%)

Figure 9: Visualizations for 15-residue dataset. Ground truth density maps are shown in blue, while predictions are shown in green. The model used to generate the ground truth electron density is shown in stick representation.

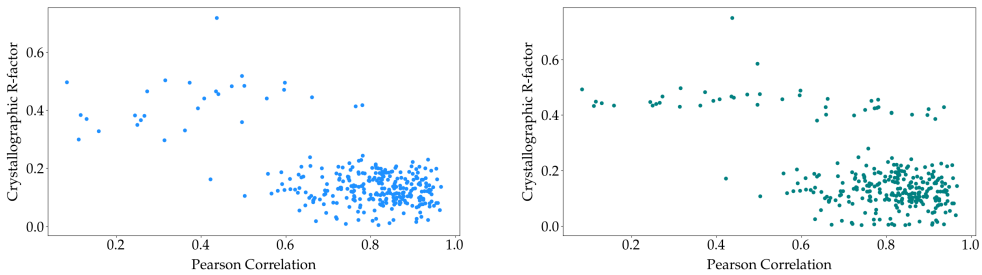


Figure 10: **Left Panel**: Scatterplot of post-refinement model R-factors, with solvent flattening applied. **Right Panel**: Scatterplot of post-refinement model R-factors, without solvent flattening applied

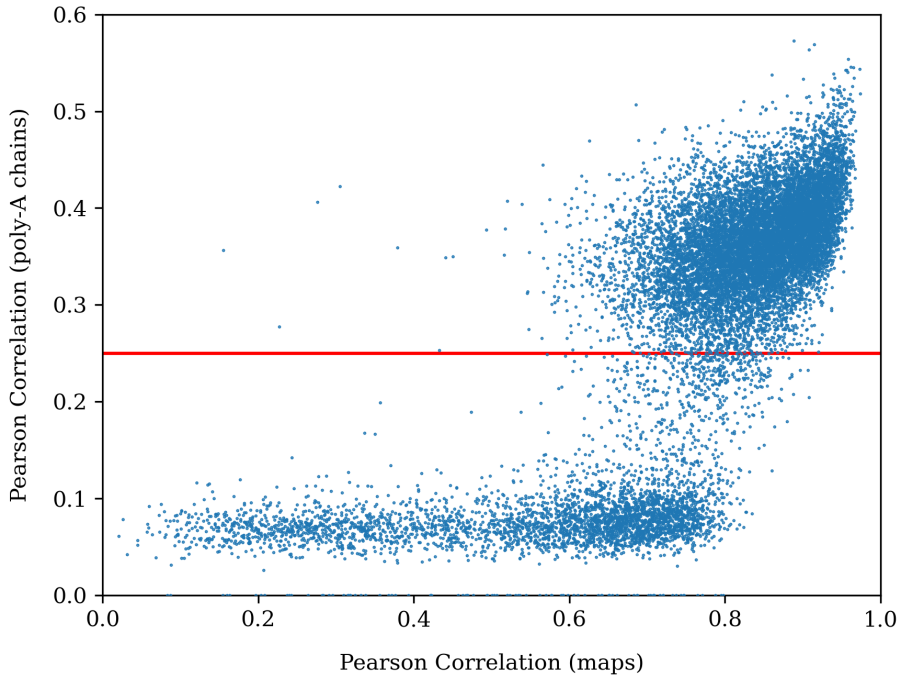


Figure 11: Scatterplot of the Pearson correlations of amplitudes of the poly-alanine chains autotraced by *shelxe* to the ground truth amplitudes vs the Pearson correlation of the predicted and ground truth maps for all 16,203 test cases