Development of Neural Networks for Biomolecular Structure Prediction with Applications to Protein Design

Rohith Krishna

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:

David Baker, Chair

Frank DiMaio

Gaurav Bhardwaj

Program Authorized to Offer Degree:

Department of Biochemistry

University of Washington

Abstract

Development of Neural Networks for Biomolecular Structure Prediction with Applications to Protein Design

Rohith Krishna

Chair of the Supervisory Committee:

David Baker

Department of Biochemistry

A grand challenge in biology is to create computational models of the interactions between abitrary biomolecular structures. In this dissertation, I describe the development of neural network models for predicting the structure of biomolecular complexes including proteins, nucleic acids, and small molecules. First, we developed a general neural network architecture for the prediction of biomolecular complexes in the Protein Data Bank (PDB). We then demonstrated the ability of this model to predict the structure of new complexes with high accuracy. Subsequently, we applied this model of native biomolecular complexes to the design of *de novo* small molecule binding proteins and enzymes. Finally, we developed a framework for development of future neural networks trained on the PDB and apply it to train several structure prediction models. To our knowledge, this dissertation represents the first efforts to develop general-purpose neural network models for biomolecular structure prediction and design.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. David Baker, for his guidance and support. His advice and feedback were invaluable in shaping this dissertation.

For most of my PhD, I worked extremely closely with Dr. Jue Wang, Dr. Frank DiMaio and Woody Ahern. Without them, this dissertation would not have been possible. Dr. Jue Wang and Dr. Frank DiMaio believed in me when no one else did, and I am deeply appreciative of their mentorship, support and friendship.

I would also like to thank my other committee members, Drs. Georg Seelig, Ning Zheng and Gaurav Bhardwaj for their unwavering support and guidance.

I was blessed to collaborate with many talented people during my PhD both at the UW and beyond. I would like to thank my colleagues (in no particular order): Jason Yim, Nate Corley, Simon Mathis, Indrek Kalvet, Pascal Sturmfels, Gyu Rie Lee, Ivan Anishchenko, Ian Humphreys, Ryan McHugh, Minkyung Baek, Doug Tischer, Saman Salike, Yakov Kipnis, Brian Coventry, Jasper Butcher, Yanjing Li, Rafi Brent, Tuscan Thompson, Meg Lunn-Halbert and many more. I thank Luki Goldschmidt for maintaining the computational infrastructure that made this dissertation possible and for his friendship.

I would like to thank my friends from high school: Andrew Low, Steven Sum, Kevin Lian and Andrew Chen for frequently visiting me in Seattle and continuing our life-long friendships. My friends from college: Kunal Desai, Justin Liu and Vincent Chiu who hosted me quite regularly when I was in the Bay Area and organized our annual trips to Scottsdale which were amazing reprieves from the rigors of graduate school. Finally, my friends in Seattle: Shreyas Gatuku, Anoushka Saxena, Arushi Patel and Harika Vedati who gave me a great

home away from home in Seattle.

My parents, Krishna and Shaila, have served as role models throughout my life and encouraged me in all my endeavors. I am indebted to them and my sister, Nithya, for their unconditional love and support. I only hope I can make them proud.

Finally, I would like to thank my partner, Sruthi, for her love, support and encouragement throughout my PhD. Without her, this dissertation would not have been possible.

Chapter 1

INTRODUCTION

Proteins rarely act alone; they form complexes with other proteins in cell signaling, interact with DNA and RNA during transcription and translation, and interact with small molecules both covalently and noncovalently during metabolism and signaling. Despite substantial recent progress in protein-only structure prediction, at the outset of this dissertation, modeling such general biomolecular assemblies remained an outstanding challenge.

The field of artificial intelligence has seen the rapid adoption of neural networks trained on large datasets, yielding state-of-the-art performance in many domains including image recognition and language translation. The principles from these successes were applied to protein structure prediction [109, 111, 129] with promising results. The breakthrough in deep learning based protein structure prediction was the development of AlphaFold [72], a deep learning model that introduced bespoke neural network architectures to predict the structure of globular proteins. AlphaFold achieves this by using *inductive biases*, which involve developing network operations that are conducive to learning the principles underlying protein structure. After the release of AlphaFold, it was unclear whether the methodological advances used in AlphaFold could be extended to the prediction of more complex biomolecular assemblies.

In this dissertation, we explore the use of inductive biases and neural network architectures to predict the structure of biomolecular assemblies including proteins, DNA, small molecules and covalent modifications. The main contribution of this thesis is setting the groundwork for the use of neural networks to model and design biomolecular assemblies which is becoming an active area of research. Tangibly, the first contribution of this dissertation is the development of the first method capable of predicting the structure of arbitrary biomolecular assemblies

with higher accuracy than existing deep learning methods. This method, called RoseTTAFold All-Atom, was then applied to many protein design tasks (described in the second and third chapter). After the publication of RoseTTAFold All-Atom, the AlphaFold team released a new method called AlphaFold3 which was also able to predict the structure of arbitrary biomolecular assemblies [2]. The last chapter of this dissertation describes the development of a new modeling framework called ModelHub which allows for training of several state of the art models using a single software package.

Chapter 2

GENERALIZED BIOMOLECULAR MODELING AND DESIGN WITH ROSETTAFOLD ALL-ATOM

This chapter is published as: Krishna, R.†, Wang, J.†, Ahern, W.†, Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G., Morey-Burows, F., Anishchenko, I., Humphreys, I., McHugh, R., Vafeados, D., Li, X., Sutherland, G., Hitchcock, A., Hunter, C., Kang, A., Brackenbrough, E., Bera, A., Baek, M., DiMaio, F., Baker, D. Generalized Biomolecular Modeling and Design with RoseTTAFold All-Atom. *Science*, 2024.

Designed the research: R.K., I.A., M.B., F.D. and D.B. Developed RFAA architecture and training regimen: R.K., J.W. and F.D. Evaluated RFAA on different structure prediction tasks: P.S., R.K., I.R.H. and R.M. Developed RFdiffusionAA: W.A. Generated designs for digoxigenin binders: P.V and G.R.L. Performed experiments for digoxigenin binders: P.V, G.R.L, D.V. and X.L. Generated designs and performed experiments for heme binders: I.K. Generated designs for bilin binders: W.A. Performed experiments for bilin binders: F.S.M-B. Contributed code and ideas: I.A., G.A.S., M.B. and F.D. Performed the crystallography experiments: A.K, E.B, A.B. Offered supervision throughout the project: D.B., A.H. and C.N.H. Wrote the manuscript: R.K., J.W., W.A and D.B.

2.1 Introduction

The deep neural networks AlphaFold2 (AF2) [72] and RoseTTAFold (RF) [11] enable high-accuracy prediction of protein structures from amino acid sequences. However, in nature, proteins rarely act alone; they form complexes with other proteins during cell signaling, interact with DNA and RNA in transcription and translation, and engage with small molecules both covalently and noncovalently during metabolism. Modeling such general biomolecular

assemblies—composed of polypeptide chains, covalently modified amino acids, nucleic acid chains, and arbitrary small molecules—remains an outstanding challenge. One common approach is to model the protein chains using AF2 or RF and then successively add the non-protein components using classical docking methods [34, 41, 42, 48, 59, 62, 63]; however, systematically evaluating and optimizing such procedures is nontrivial. RF has been extended to model both proteins and nucleic acids by increasing the residue alphabet to 28 (20 amino acids, 4 DNA bases, and 4 RNA bases) with RoseTTAFold nucleic acid (RFNA) [12], but general biomolecular system modeling remains a greater challenge due to the diversity of possible small molecule components. An approach capable of accurately predicting the three-dimensional structures of biomolecular assemblies starting only from knowledge of the constituent molecules (and not their 3D structures) would have broad impact on structural biology and drug discovery, opening the door to deep learning–based design of protein–small molecule assemblies. We set out to develop a structure prediction method capable of generating 3D coordinates for all atoms of a biological unit, including proteins, nucleic acids, small molecules, metals, and chemical modifications (2.1A). The first obstacle we faced in taking on the broader challenge of generalized biomolecular system modeling was how to represent the components. Existing protein structure prediction networks represent proteins as linear chains of amino acids, and this representation can be readily extended to nucleic acids. However, many of the small molecules that proteins interact with are not polymers, and it is unclear how to model them as a linear sequence. A natural way to represent the bonded structure of small molecules is as graphs whose nodes are atoms and whose edges represent bond connectivity. This graph representation is not suitable for proteins as they contain many thousands of atoms; hence, modeling whole proteins at the atomic level is computationally intractable. To overcome this limitation, we sought to combine a sequence-based description of biopolymers (proteins and nucleic acids) with an atomic graph representation of small molecules and protein covalent modifications.

2.2 Generalizing Structure Prediction to All Biomolecules

We modeled the network architecture after the RoseTTAFold2 (RF2) protein structure prediction network, which accepts 1D sequence information, 2D pairwise distance information

from homologous templates, and 3D coordinate information and iteratively improves predicted structures through many hidden layers[10]. We retain the representations of protein and nucleic acid chains from RF2 and represent arbitrary small molecules, covalent modifications and unnatural amino acids as atom-bond graphs. To the 1D track, we input the chemical element type of each non-polymer atom; to the 2D track, the chemical bonds between atoms; and to the 3D track, information on chirality [whether chiral centers are (R) or (S)]. For the 1D track, we supplement the 20 residue and eight nucleic acid base representation in RFNA with 46 new element type tokens representing the most common element types found in the Protein Data Bank (PDB) (Table S5). For the 2D track atom-bond embedding, we encode pairwise information about whether bonds between pairs of atoms are single, double, triple, or aromatic bonds. These features are linearly embedded and summed with the initial pair features at the beginning of every recycle of the network, allowing the network to learn about bond lengths, angles, and planarity. Since the 1D and 2D representations in the network are invariant to reflections, we encode stereochemistry information in the third track by specifying the sign of angles between the atoms surrounding each chiral center (Fig S1); at each block in the 3D track the gradient of the deviation of the actual angles from the ideal values (with respect to the current coordinates) is computed and provided as an input feature to the subsequent block (2.1B).

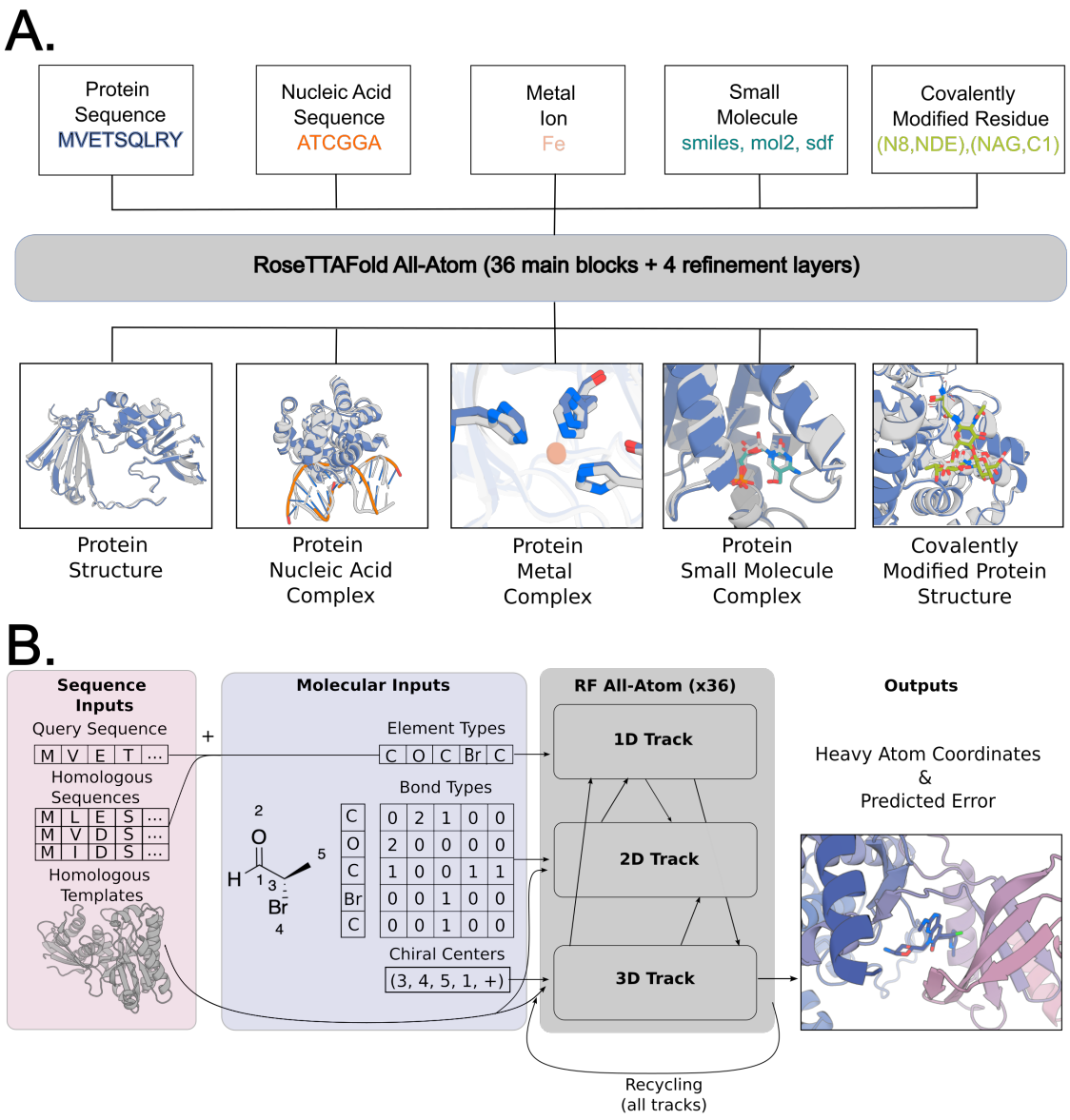


Figure 2.1: General biomolecular modeling with RoseTTAFold All-Atom A) RFAA takes input information about the molecular composition of the biomolecular assembly to be modeled, including protein amino acid and nucleic acid base sequences, metal ions, small molecule bonded structure, and covalent bonds between small molecules and proteins. B) Processing of molecular input information. Small molecule information is parsed into element types (46 possible types), bond types, and chiral centers. Covalent bonds between proteins and small molecules are provided as a separate token in the bond adjacency matrix. The three-track architecture mixes 1D, 2D, and 3D information and predicts all-atom coordinates and model confidence.

Unlike proteins and nucleic acid sequences, molecular graphs are permutation invariant, and hence, the network should make the same prediction irrespective of small molecule element token order. In AF2 and RF2, the sequence order of amino acids and bases is represented by a relative position encoding; for atoms, we omit such an encoding and leverage the permutation invariance of the network attention mechanisms. We also modify the coordinate updates: in AF2 and RF, protein residues are represented by the coordinates of the Cα and the orientation of the N-Cα-C rigid frame (or the P coordinate and the OP1-P-OP2 frame orientation in RFNA) and along the 3D track the network generates rotational updates to each frame orientation and translational updates to each coordinate. To generalize this representation in RFAA, heavy atom coordinates are added to the 3D track and move independently based only on a predicted translational update to their position. Thus, immediately after input, the full system is represented as a disconnected gas of amino acid residues, nucleic acid bases, and freely moving atoms, which is successively transformed through the many blocks of the network into physically plausible assembly structures.

For the loss function to guide parameter optimization, we develop an all-atom version of the Frame Aligned Point Error (FAPE) loss introduced in AF2 by defining coordinate frames for each atom in an arbitrary molecule based on the identities of its bonded neighbors and, as with residue-based FAPE, successively aligning each coordinate frame and computing the coordinate error on the surrounding atoms (Figure 2A; for greater sensitivity to small molecule geometry, we upweight contributions involving atoms; see Supplemental Methods). In addition to atomic coordinates, the network predicts atom- and residue-wise confidence (pLDDT) and pairwise confidence (PAE) metrics to enable users to identify high-quality predictions. A full description of the RFAA architecture is provided in the Supplemental Methods.

2.3 Training RFAA

We curated a protein-biomolecule dataset from the PDB including protein-small molecule, protein-metal, and covalently modified protein complexes, filtering out common solvents and crystallization additives. Following clustering (30% sequence identity) to avoid bias

towards overrepresented structures, we obtained 121,800 protein-small molecule structures in 5,662 clusters, 112,546 protein-metal complexes in 5,324 clusters, and 12,689 structures with covalently modified amino acids in 1,099 clusters for training. To help the network learn the general properties of small molecules rather than features specific to the molecules in the PDB, we supplemented the training set with small molecule crystal structures from the Cambridge Structural Database[54]. Each training example is sampled uniformly from the set of organic non-polymeric molecules, and the network predicts the coordinates for the asymmetric unit given atomic graph information. To further help the network learn about general atomic interactions, we take advantage of the commonalities between atomic interactions within proteins and many of the atomic interactions between proteins and small molecules and augment the training data by inputting portions of proteins as atoms rather than residues (a process we term atomization). We atomize randomly selected subsets of three to five contiguous residues by deleting the sequence and template features and providing instead atom, bond, and chirality information for the atoms in those residues (an alanine would be replaced by five atom tokens, one for each heavy atom). Since the atoms are still part of the polypeptide chain, we provide the relative position of the atom tokens with respect to the other residue tokens by adding an extra bond token that corresponds to an "atom-to-residue" bond and develop a positional encoding to account for atom-residue bonds (Supplemental Methods). To increase prediction accuracy on biological polymers, we train the network on protein monomer, protein complex, and protein-nucleic acid complex examples as previously described[10, 12]. All examples were cropped to have 256 tokens during the initial stages of training and 375 tokens during fine-tuning. The progress of training was monitored using independent validation sets consisting of 10% of the protein sequence clusters (see Table S4).

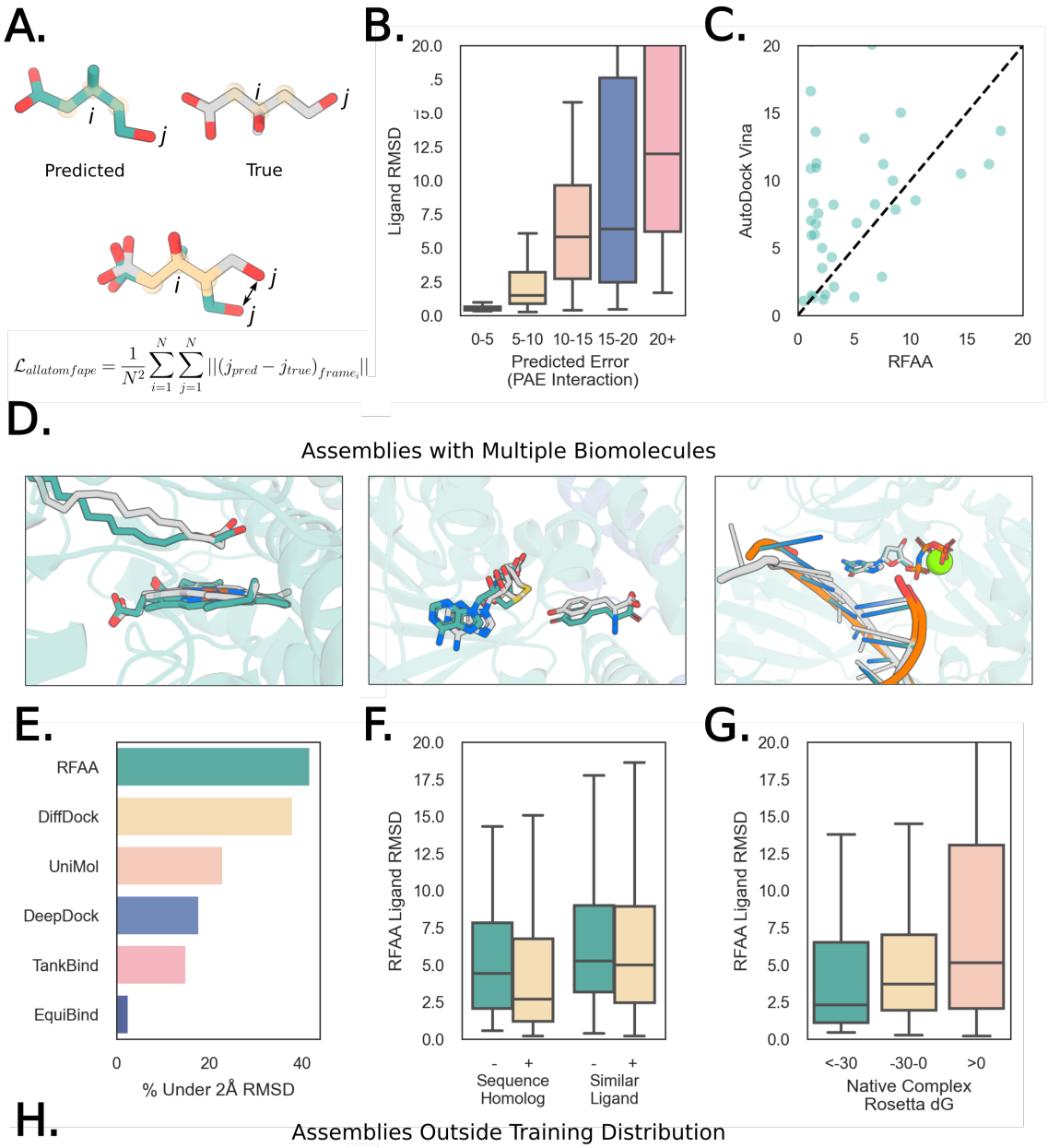

A.
Predicted
True
$\mathcal{L}_{allatomfape} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}||(j_{pred} - j_{true})_{frame_i}||$
B.
Ligand RMSD
Predicted Error
(PAE Interaction)
0-5
5-10
10-15
15-20
20+
C.
AutoDock Vina
RFAA
D.
Assemblies with Multiple Biomolecules
E.
RFAA
DiffDock
UniMol
DeepDock
TankBind
EquiBind
% Under 2Å RMSD
F.
RFAA Ligand RMSD
Sequence Homolog
Similar Ligand
G.
RFAA Ligand RMSD
Native Complex Rosetta dG
H.
Assemblies Outside Training Distribution
Closest Protein Seq in Training: 25%
Closest Ligand In Training: 1.0
Ligand RMSD: 0.38
Closest Protein Seq in Training: 38%
Closest Ligand In Training: 0.41
Ligand RMSD: 0.89
Closest Protein Seq in Training: 23%
Closest Ligand In Training: 0.46
Ligand RMSD: 1.20

Figure 2.1 *(previous page)*: RoseTTAFold All-Atom can accurately predict protein-small molecule complex structures. All panels: Predicted protein structure (aligned to native): transparent teal, predicted ligand conformation: teal, native ligand conformation: gray. All boxplots cut off at 20 Å for clarity. A) Every "atom" node is assigned a local coordinate frame based on the identities of its neighbors. To compute the main loss in the network, we align each atom's coordinate frame in the predicted and true structures and measure the error over all the other atoms. B) Model accuracy correlates with error predictions. Computed for CAMEO targets (05/20/23-7/29/23; 261 protein-small molecule interfaces). Ligand RMSD was computed by CAMEO organizers. C) RFAA outperforms AutoDock Vina on CAMEO targets (Week 8/12/23-09/02/23; 149 protein-small molecule interfaces). Both servers have to model the protein, find pockets for all ligands present in the solved structure, and the correct docks for all ligands. Ligand RMSD for both servers was computed by CAMEO organizers, AutoDock Vina server set up by CAMEO organizers. D) Three examples of successful predictions with multiple biomolecules. From left to right: fatty acid decarboxylase (PDB ID: 8d8p; Seq ID: 34%; from CAMEO) with a heme cofactor and a lipid substrate, a dimeric tyrosine methyltransferase (PDB ID: 7ux8; Seq ID: 28%; CASP15 Target: T1124) with an S-adenosyl homocysteine and tyrosine interaction and a DNA polymerase (PDB ID: 7u7w; Seq ID: 100%) bound to DNA, a nucleotide and a metal ion[29, 101, 137]. E) Comparison to other deep learning-based docking methods. In this case, each method was applied in their respective training regime. For RFAA this means only having sequence and minimal atomic graph inputs, whereas for other methods this involves providing the bound crystal structure. Ligand RMSD was computed using PoseBusters suite, and a single example present in our training set was removed for all methods in comparison. F) Comparison of RFAA predictions on recently solved PDBs that are novel compared to the training set (Homolog $<$1 BLAST e-value, Similar Ligand $>$0.5 Tanimoto Similarity). Each set is clustered based on sequence/ligand similarity, and a random cluster representative is chosen for each. G) Comparison of RFAA prediction accuracy to Rosetta ΔG energy estimates for the native complex (over 940 cases that were successfully processed by Rosetta). RFAA makes more accurate predictions for native complexes with low Rosetta energy. H) Three examples of successful predictions with low similarity to the training set. From left to right: G protein-coupled S1P receptor (PDB ID: 7ew1; Seq ID: 31%), complex of DLK bound to an inhibitor (PDB ID: 8ous; Seq ID: 39%), a Renilla luciferase bound to an azacoelenterazine (non-native substrate; PDB ID: 7qxr; Seq ID: 23%).[78, 110, 145].

Unlike previous protein-only deep learning architectures[46, 83, 139], RFAA can model full biomolecular systems. In the following sections, we describe the performance of RFAA on different structure modeling tasks. We adopted the philosophy that a single model trained on all available data over all modalities would have the greatest ability to generalize and be more accessible than a series of models specialized for specific problems.

2.4 Predicting Protein-Small Molecule Complexes

To enable blind testing of RFAA prediction performance, we enrolled an RFAA server in the blind CAMEO ligand docking evaluation, which carries out predictions on all structures submitted to the PDB each week with each enrolled server and evaluates their performance[56–58]. These structures can have multiple protein chains, ligands, and metal ions (for further results on metal ions, see Figure S2). Of the CAMEO targets, 43% are predicted confidently by RFAA (PAE Interaction < 10), and 77% of those high-confidence structures are quite accurate, with < 2 Å ligand RMSD (2.1B). One of the other servers is an implementation of a leading non-deep learning protein small molecule docking method AutoDock Vina by the CAMEO organizers that predicts the protein structure by homology modeling[18, 22, 117, 125, 131], runs AutoDock to dock the small molecules, and ranks the poses using the Vina scoring function[42, 131]. RFAA consistently outperformed the other servers in CAMEO on protein-small molecule modeling; for example, on cases modeled by both the RFAA and the AutoDock Vina servers, RFAA models 32% of cases successfully (< 2 Å ligand RMSD) compared to 8% for the Vina server (2.1C; the Vina performance by an expert would likely be considerably improved because of the complexities of fully automatic multiple step modeling pipelines). The most common RFAA failure mode is the placement of small molecules in the correct pockets but not in the correct orientation (Figure S3; for further exploration of failure modes, see Supplemental Methods).

There were no other deep learning docking methods[34, 81, 87, 100, 118, 149] enrolled in CAMEO, but we can instead compare performance on a set of PDB structures that were solved after our training set date cutoff [25] (most earlier deep learning based docking tools have focused on the "bound" docking problem where the crystal structure of the target (including sidechains) are provided, and hence are less well suited to CAMEO). On this benchmark, RFAA predicts 42% of complexes successfully compared to DiffDock, which predicts 38% of complexes successfully (2.1D; RFAA predicts the protein backbone and side chains in addition to the small molecule dock, whereas DiffDock receives the crystal structure of the protein from the bound complex as input). In cases where both the bound protein

structure and the pocket residues are provided, physics-based methods such as AutoDock Vina outperform RFAA (52% vs 42%), which has the much harder task of predicting both the protein backbone and sidechain details and the dock from sequence alone (Figure S4A).

To further benchmark the network, we assembled a dataset of recent PDB entries with small molecules bound that were deposited after the cutoff date for our training set and predicted full structure models for all 5,421 complexes (1,529 protein sequence clusters at 30% sequence identity). The network performs better for clusters with overlap with the training set, but also generates accurate predictions for proteins with low (BLAST e-value > 1) sequence similarity to the training set (35% vs. 24% success rate, respectively; 2.1F). We observe a similar pattern for ligand clusters (across 1,310 ligand clusters); whereas the network makes more accurate predictions for ligands seen in training, it also can make accurate predictions on ligands that are not similar to those in training ($<$0.5 Tanimoto similarity; 19% vs. 14% success rate) (2.1F). In cases where RFAA predicts ligand placement with high confidence and RF2 has high confidence (PAE Interaction $<$10 and pLDDT $>$0.8 respectively), RFAA makes higher accuracy protein structure predictions than RF2 (Fig S4A), indicating that training with ligand context can improve overall protein prediction accuracy. Some examples of shifts predicted by RFAA but not by RF2 include domain movements, subtle backbone movements, and flipping of side chain rotamers to accommodate the ligand in the pocket (Figure S4B-C).

Unlike previous methods, RFAA is able to jointly predict interactions between proteins and multiple non-protein ligands in a single forward pass. 2.1D shows three examples of recently solved structures with three or more components for which RFAA predictions had $<$2 Å ligand RMSD (when the proteins are aligned). There are homologous complexes in the training set so these are not de novo predictions, but they do demonstrate that RFAA can learn the multicomponent assembly prediction task. The right panel shows a prediction for DNA polymerase [29](PDB ID: 7u7w) with a bound DNA, non-hydrolyzable guanine triphosphate and magnesium ion; the network received no examples of higher order assemblies containing proteins with both small molecules and nucleic acids during training, but is likely synthesizing information from multiple related binary complexes that are in the training set.

To assess whether the network can distinguish compounds known to bind from related compounds, we compared protein-small molecule complex predictions for the PoseBusters dataset for the compound known to bind and decoy molecules including small molecules with the highest Tanimoto similarity in the dataset. In 75.1% of cases the PAE interaction metric of the "decoy" complex was higher (indicating lower confidence) than the native complex (Figure S7). Direct optimization on this discrimination task would likely further improve performance.

To determine the extent to which the network is reasoning over the detailed structure of protein-small molecule interactions, we investigated the correlation between prediction accuracy and the interaction energy computed by a molecular force field. We found that predictions for protein-small molecule complexes in our recent PDB set with lower computed binding energies (by Rosetta ΔG)[7, 94] were more accurate (2.1G; 50%, 25%, and 22% success rates for <-30, -30-0, and >0 Rosetta Energy Units, respectively) suggesting the network considers the detailed interactions between the protein and small molecule (although reasoning over these interactions very differently than human designed force fields).

2.5 Predicting Structures of Covalent Modifications to Proteins

Many essential protein functions, such as receptor signaling, immune evasion, and enzyme activity, involve covalent modifications of amino acid side chains with sugars, phosphates, lipids, and other molecules[13, 80, 102, 106]. RFAA models such modifications by treating the residue and chemical moiety as atoms (with the corresponding covalent bond to the atom token in the residue) and the rest of the protein structure as residues (2.1A). Unnatural amino acids can be modeled in the same way.

We benchmarked the performance of RFAA on covalent modification structure prediction on 931 recent entries in the PDB (post-May, 2020), and found that the network made accurate predictions (Modification RMSD<2.5 Å) in 46% of cases (where Modification RMSD is defined as RMSD of the modified residue and chemical modification when the rest of the protein is aligned). As in the protein-small molecule complex case, confident predictions tend to be more accurate: 60% of structures are predicted with high confidence (PAE

Interaction <10), and 63% of those predictions are accurate (<2.5 Å modification RMSD) (2.1B). Although the network makes slightly more accurate predictions on cases with sequence similarity (>25% identity) to proteins in the training set, there are still many cases (27.5%) that do not have sequence overlap to the training set that are predicted with high accuracy (2.1C). RFAA models interactions with covalently bound cofactors and covalently bound drugs with median RMSDs of 0.99 Å and 2.8 Å respectively (2.1D-E).

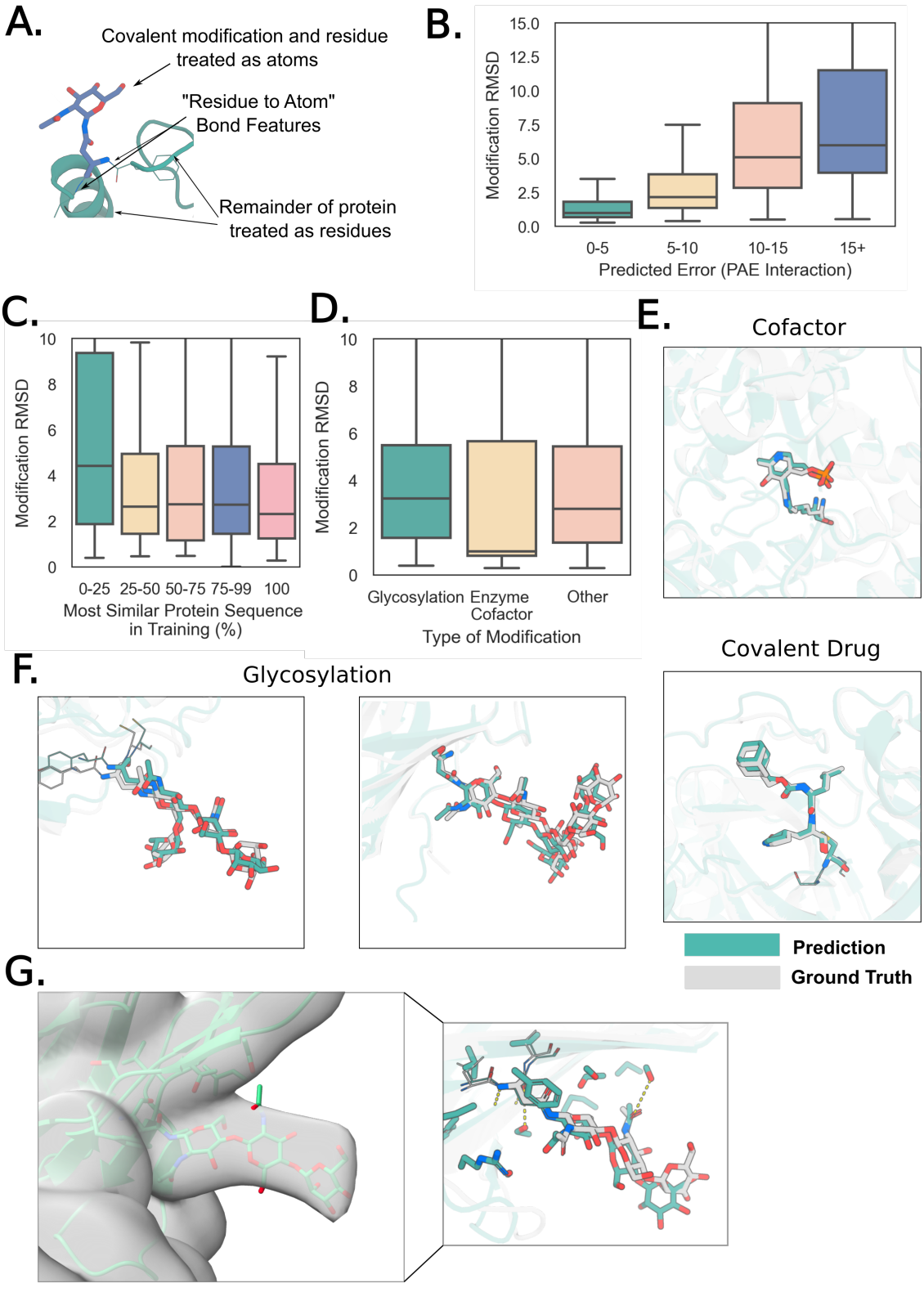

A.
Covalent modification and residue treated as atoms
"Residue to Atom" Bond Features
Remainder of protein treated as residues
B.
Modification RMSD
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0-5
5-10
10-15
15+
Predicted Error (PAE Interaction)
C.
Modification RMSD
10
8
6
4
2
0
0-25
25-50
50-75
75-99
100
Most Similar Protein Sequence in Training (%)
D.
Modification RMSD
10
8
6
4
2
0
Glycosylation
Enzyme Cofactor
Other
Type of Modification
E.
Cofactor
Covalent Drug
F.
Glycosylation
Prediction
Ground Truth
G.

Figure 2.1 *(previous page)*: Accurate prediction of protein covalent modifications. All panels: transparent teal: predicted protein structure, transparent gray: native structure, teal: predicted covalent modification, gray: native covalent modification. A) Schematic describing how RFAA models covalent modifications to proteins. The chemical moiety that modifies the residue and the residue are modeled as atom nodes, and the rest of the protein is modeled as residues (with MSA and template inputs). B) Model accuracy correlates with predicted error on a set of 938 recently solved structures with covalent modifications. Modification RMSD is computed by aligning the protein structure within 10 Åand computing RMSD over the modified residue and chemical modification. Boxplot cut off at 15 Åfor clarity. C) Comparison of sequence identity to training set and model accuracy. Models are generally accurate even with low sequence homology to the training set. D) Comparison of model accuracy for different types of covalent modifications. E) Top: Example of successfully predicted covalently linked enzyme cofactor (PDB ID: 7p3t; Seq ID: 28%), which is a structure of a (R)-selective amine transaminase. Bottom: example of a covalently bound drug candidate (PDB ID: 7ti1, Seq ID: 27%), which is a β-lactamase enzyme bound to cyclic boronic acid inhibitor[75, 105]. F) Accurate predictions of glycans on the N-acetylglucosamine-1-phosphotransferase (GNPT) gamma subunit (PDB ID: 7s69; No BLAST hits), human sperm TMEM ectodomain (PDB ID: 7ux0; Seq ID: 26%)[52, 119].

Prediction of glycan structure has applications in therapeutics, vaccines, and diagnostics[5, 71, 136]. RFAA can accurately model carbohydrate groups introduced by glycosylations with a median RMSD over our test set of 3.2 Å (2.1D). RFAA successfully predicts glycan conformations on the N-acetylglucosamine-1-phosphotransferase (GNPT) gamma subunit (PDB ID: 7s69), and human sperm TMEM ectodomain (PDB ID: 7ux0), which have low sequence homology (<30%) to the RFAA training set (2.1F) and have multiple monosaccharides and different branching patterns[52, 119]. RFAA is not simply learning how structure building programs model glycans as the predictions match the experimental density maps (Fig S8C). The network is able to make accurate predictions of glycan interactions even when the sequences were distant from the sequences in the training set, and on glycans with chains up to seven monosaccharides (Figure S8).

It is difficult to compare to other methods because, to our knowledge, previous deep learning-based tools do not model covalent modifications to proteins. Accurate and robust modeling of covalent modifications in predicted structures should contribute to the understanding of

biological function and mechanism.

2.6 De Novo Small Molecule Binder Design

Previous work on small molecule binding protein design has involved docking molecules into large sets of native or expert-curated protein scaffold structures[21, 98]. Diffusion based methods can generate proteins in the context of a protein target that bind with considerable affinity and specificity[132] and can be trained to explicitly condition on structural features[92]. However, current deep learning based generative approaches do not explicitly model protein-ligand interactions, so they are not directly applicable to the small molecular binder design problem (in RFdiffusion, a heuristic attractive-repulsive potential encouraged the formation of pockets with shape complementarity to a target molecule, but the approach was unable to model the details of protein-small molecule interactions [132]). A general method that can generate protein structures around small molecules and other non-protein targets to maximize favorable interactions could be broadly useful.

We reasoned that RFAA could enable protein design in the context of non-protein biomolecules following fine-tuning on structure denoising. We developed a diffusion model, RFdiffusion All-Atom (RFdiffusionAA), by training a denoising diffusion probabilistic model (DDPM) initialized with the RFAA structure-prediction weights to denoise corrupted protein structures conditioned on small molecules and other biomolecular context (2.1A). Input structures from the protein-small molecule dataset described above were noised through progressive addition of 3D Gaussian noise to the Cα coordinates and Brownian motion on the manifold of rotations, and the model was trained to predict the denoised structures. In contrast to training for the unconditional generation problem and incorporating conditional information through forms of guidance[70, 138], we train an explicitly conditional model that learns the distribution of proteins conditioned on biomolecular substructure. To enable the inclusion of specific protein functional motifs when desired, we also train the network to scaffold a variety of discontiguous protein motifs both in the presence and absence of small molecules. To generate proteins, we initialize a Gaussian distribution of residue frames with randomized rotations around a fixed small molecule motif; at each denoising step t, we predict the fully

denoised X_0 state and then update all residue coordinates and orientations by taking a step towards this conformation while adding noise to match the distribution for X_{t-1}. As with RFdiffusion, we investigated the use of auxiliary potentials to influence trajectories to make more contacts between small molecules and binders, but found these unnecessary (see Figure S10C).

We evaluated RFdiffusionAA in silico by generating protein structures in the context of four diverse small molecules. Starting from random residue distributions surrounding each of the small molecules, iterative denoising yielded coherent protein backbones with pockets complementary to the small molecule target. Following sequence design using LigandMPNN [38], Rosetta GALigandDock [94] energy calculations were used to evaluate the protein-small molecule interface and AF2 predictions to evaluate the extent the sequence encodes the designed structure [132]. The computed binding energies of RFdiffusionAA designs are far better ($p<1.56E-12$) than those obtained using a heuristic attractive/repulsive potential with protein-only RFdiffusion (Figure S10C). AF2 structure predictions had backbone RMSD < 2 Å to the RFdiffusionAA design models in all cases (Figure S10C). For each small molecule, RFdiffusionAA generates diverse protein structural solutions to the binding problem that differ from native binders to these ligands (Figure S11, Figure S12).

2.7 Experimental Characterization of Designed Binders

To experimentally evaluate RFdiffusionAA across a range of design scenarios, we designed binders for three diverse small molecules: one with no protein motif included in the design parameters, one with a single residue protein motif, and one with a four residue protein motif (Fig. 4). The proteins were produced in E. coli, and ligand binding was measured experimentally.

Digoxigenin (DIG) is the aglycone of digoxin, a small molecule used to treat heart diseases with a narrow therapeutic window[40], and digoxigenin-binding proteins could help reduce toxicity [47]. Previous attempts to design digoxigenin-binding proteins relied on protein scaffolds with experimentally determined structures and prespecified binding pockets and interacting motifs[122]. We used RFdiffusionAA to design digoxigenin-binding backbones

without any prior assumption about the protein-ligand interface or backbone structure (2.1A). Sequences were obtained using LigandMPNN and Rosetta FastRelax [36] and 4,416 designs were selected based on consistency with AF2 predictions and Rosetta metrics (Supplemental Methods). Experimental characterization identified several DIG-binding proteins (Figures S29-30, Supplemental Methods); the highest affinity binder has a 343 nM Kd for free digoxigenin (measured by isothermal titration calorimetry, 2.1B) and is stable at temperatures up to 95°C.

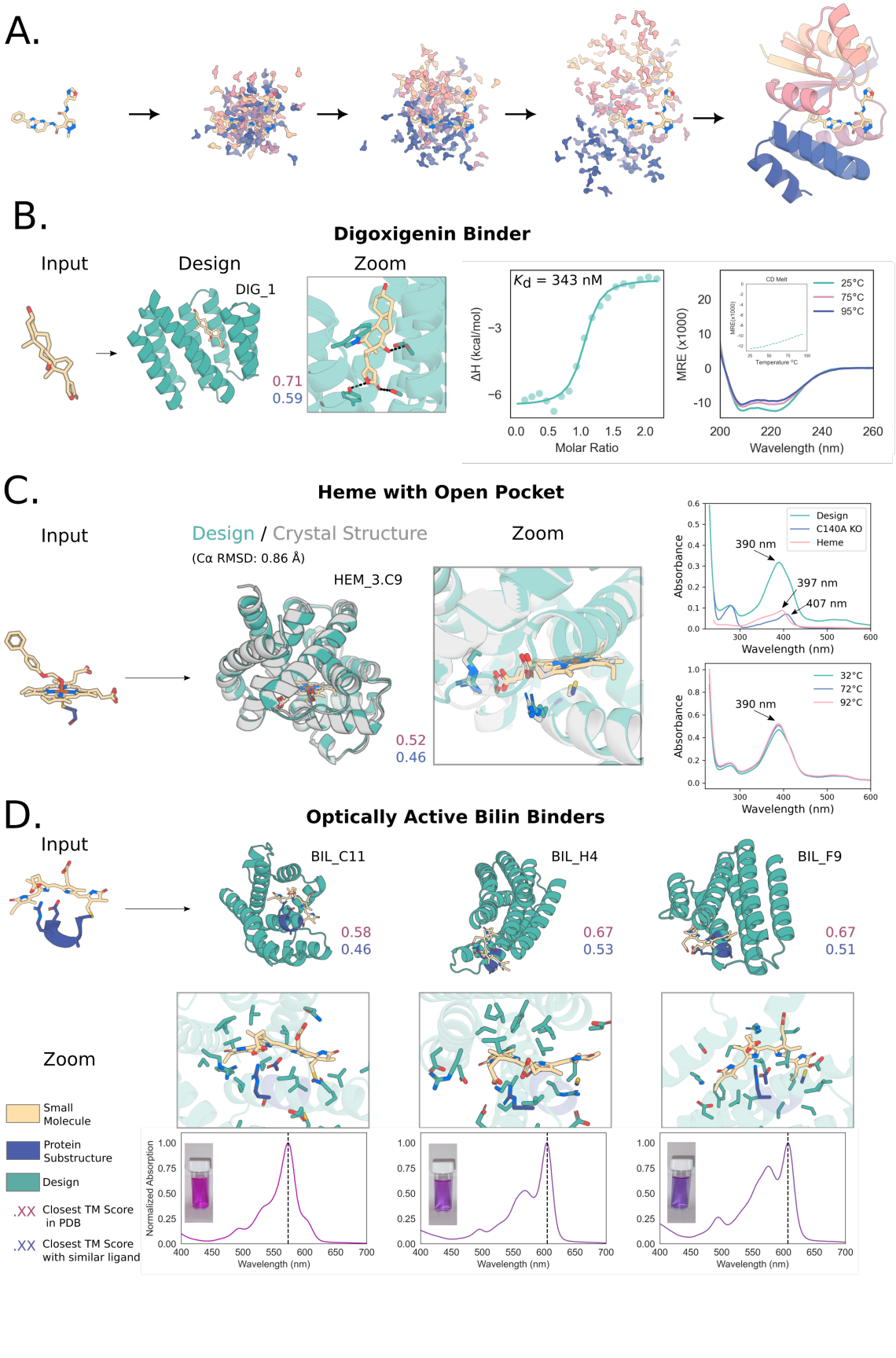

A.
B.
Digoxigenin Binder
Input
Design
Zoom
DIG_1
0.71
0.59
Kd = 343 nM
ΔH (kcal/mol)
Molar Ratio
CD Melt
25°C
75°C
95°C
MRE (x1000)
Wavelength (nm)
C.
Heme with Open Pocket
Input
Design / Crystal Structure
(Cα RMSD: 0.86 Å)
Zoom
HEM_3.C9
0.52
0.46
Design
C140A KO
Heme
390 nm
397 nm
407 nm
Absorbance
Wavelength (nm)
32°C
72°C
92°C
D.
Optically Active Bilin Binders
Input
BIL_C11
0.58
0.46
BIL_H4
0.67
0.53
BIL_F9
0.67
0.51
Zoom
Small Molecule
Protein Substructure
Design
.XX Closest TM Score in PDB
.XX Closest TM Score with similar ligand
Normalized Absorption
Wavelength (nm)

Figure 2.1 *(previous page)*: Experimental characterization of RFdiffusionAA designed binders. All panels: input ligand shown in yellow, input protein motif shown in blue, and diffused protein shown in teal. Purple text: Closest TM Score to any protein in the training set, Blue text: Closest TM Score to any protein with a similar ligand bound in the training set (Tanimoto <0.5). A) Schematic depicting the random initialization of residues surrounding a small molecule and progressive denoising by RfdiffusionAA. B) Characterization of dioxigenin binder design. (From left to right) Input motif to RFdiffusionAA, designed protein, zoom in view of binding site sidechains. Isothermal Calorimetry (ITC) measuring binding affinity (Kd = 343 nM), CD trace (26 μM protein concentration; inlay CD Melt showing intensity at 220 nm across a broad range of temperatures). C) Characterization of heme binding designs. (From left to right) Input motif to RFdiffusionAA, designed protein aligned to its crystal structure (PDB ID: 8vc8); zoom in view of binding site; (top) UV-Vis spectra of designed protein matches expected spectra for penta-coordinated heme and mutating cysteine to alanine abolishes binding; (bottom) designed protein retains heme binding at temperatures up to 90°C. D) Characterization of bilin binding designs. (Row 1, left to right) Input motif to RFdiffusionAA, three designs with different predicted structural topologies. (Row 2, left to right) Zoom in view of binding sites for each design. (Row 3, left to right) Normalized absorption spectra for the three designs shown. Designs have a range of maximum absorption wavelengths and hence different colors in solution (inset).

Heme is a cofactor for a wide range of oxidation reactions and oxygen transport (cytochrome P450 and hemoglobin are two notable examples), with catalytic function enabled by pentacoordinate iron binding and an open substrate pocket[67, 99]. Designed heme-binding proteins with these features have considerable potential as a platform for the development of new enzymes[74]. We diffused proteins around heme with the central iron coordinated by a cysteine and placeholder molecule just above the porphyrin ring to keep the axial heme binding site open for potential substrate molecules. Of 168 designs selected based on AF2 predicted confidence (pLDDT), backbone RMSD to design, and RMSD of the predicted cysteine rotamer to the design, 135 were well expressed in E. coli, and 90 had UV/Vis spectra consistent with Cys-bound heme (as judged by the Soret maximum wavelength after in vitro heme loading)[113]. We further purified 40 of the designs and found that 33 were monomeric and retained heme-binding through size exclusion chromatography (SEC). For 26 of the designs, we mutated the putative heme-coordinating cysteine residue to alanine which led to a notable change in the Soret features in all cases (2.1; Figure S13-16). Twenty designs

exhibit high thermostability, retaining their heme binding at temperatures above 85°C, and do not unfold at temperatures up to 95°C (2.1C and Figure S13-16). We solved the crystal structure of heme-loaded design HEM_3.C9 to 1.8 Å resolution (PDB ID: 8vc8) and found it to closely match the design model (0.86 Å Cα RMSD). The crystal structure verifies that heme is bound through Cys-ligation in a pentacoordinate fashion with an open distal pocket (in agreement with spectroscopic data) and is further held in place with hydrogen bonds to two arginines, as designed (Figure S17).

Bilins are brilliantly colored pigments that play important roles across diverse biological kingdoms. When bilins are constrained by protein scaffolds, such as phycobiliproteins in the megadalton phycobilisome antenna complexes of cyanobacteria and some algae [4], their absorption features narrow, their extinction coefficients increase, and their fluorescence is dramatically enhanced. We sampled diffusion trajectories conditioned on the structure of a bilin molecule attached to a four residue peptide corresponding to a motif recognized by the CpcEF bilin lyase [90, 148]. We evaluated 94 designs with a whole cell screen using phycoerythrobilin (PEB) as the chromophore and identified nine proteins dissimilar to each other and to CpcA (Figure S18A) that bind bilin based on pigmentation or fluorescence (a 9.6% hit rate). We purified three designs - BIL_C11, BIL_H4, and BIL_F9 - with absorption maxima at 573, 605, and 607 nm compared to 557 nm for the CpcA-PEB (Figure 5C, S8B; the extent of red shifting correlates with computed electrostatic potential around the chromophore (Figure S19)). Conformationally restricted bilins typically display higher fluorescence yields, absolute fluorescence yields for the BIL_C11, BIL_H4, and BIL_F9 designs are 38%, 11% and 25%, respectively, based on an earlier determination of the absolute fluorescence quantum yield for CpcA-PEB of 67% [15] (Figure S18C). These values are much higher than obtained previously with maquette scaffolds ($F\Phi$ values of $2-3\%$), which displayed limited bilin incorporation and less pronounced spectral enhancements [89]. The strong coloration, absorption and emission for these designs were absent from control E. coli strains that synthesize only the PEB bilin and the CpcE/F lyase, or PEB, CpcE/F and maltose binding protein (Figure S20). The 34/30 nm range in absorption/emission covered by just one design round using a single chromophore raises the exciting prospect of tailoring

the spectral profiles of designed biliproteins by manipulating the conformational flexibility of the bilin and the protein microenvironment. De novo designed antenna complexes could harvest light over a wider range of the UV-visible spectrum to enhance photosynthetic energy capture and conversion [60], and fluorescent reporter probes with tunable excitation/emission maxima would be useful biochemical tools.

The experimental validation of digoxigenin, heme and bilin binding proteins demonstrates that RFdiffusionAA can readily generate novel proteins with custom binding pockets for diverse small molecules. Unlike prior methods that rely on redesigning existing scaffolds, RFdiffusionAA builds proteins from scratch around the target compound, resulting in high shape-complementary in the binding pockets and reducing the need for expert knowledge. The ability of RFdiffusionAA to generalize is highlighted by the sequence and structural dissimilarity between the designs and proteins in the PDB that bind related molecules (related meaning Tanimoto similarity > 0.5); the most similar protein in the PDB that binds a related molecule has a TMscore of 0.59 for the highest affinity digoxigenin binder, less than 0.62 for all the characterized heme binders, and less than 0.52 for the bilin binders (Figure S21). In all cases there is no detectable sequence similarity to any known protein.

2.8 Discussion

RoseTTAFold All-Atom (RFAA) demonstrates that a single neural network can be trained to accurately model a wide range of general biomolecular assemblies containing a wide diversity of non-protein components. RFAA can make high-accuracy predictions on protein-small molecule complexes, with 32% of CAMEO targets predicted under 2 Å RMSD, and for covalent modifications to proteins, predicting 46% of recently solved covalent modifications under 2.5 Å RMSD, and generate accurate models for complexes of proteins with two or more non-protein molecules (small molecules, metals, nucleic acids, etc.). Training on more extensive datasets will likely be necessary to generate consistently accurate predictions for new protein-small molecule complexes on par with the accuracy deep networks can achieve on protein systems alone. These new prediction capabilities do not come at the expense of performance on the classic protein structure prediction problem: RFAA achieves similar

protein structure prediction accuracy as AF2 (median GDT of 85 vs. 86) and protein-nucleic acid complex accuracy as RFNA (median allatom-LDDT of 0.74 vs. 0.78) (Figure S22).

Our prediction and design results suggest that RFAA has learned detailed features of protein-small molecule complexes. First, the network is able to make high-accuracy predictions for protein sequences and ligands that differ considerably from those in the training dataset (2.1F, 3C), and prediction accuracy is higher for complexes with more favorable computed interaction energies using the Rosetta physically based model (2.1G). Second, our RFdiffusionAA-generated bilin, heme, and digoxigenin binders have very different structures than proteins that bind these compounds in the PDB. RFAA should be immediately useful for modeling protein-small molecule complexes, in particular multicomponent biomolecular assemblies for which there are few or no alternative methods available, and for designing small molecule binding proteins and sensors.

2.9 Supplemental Figures

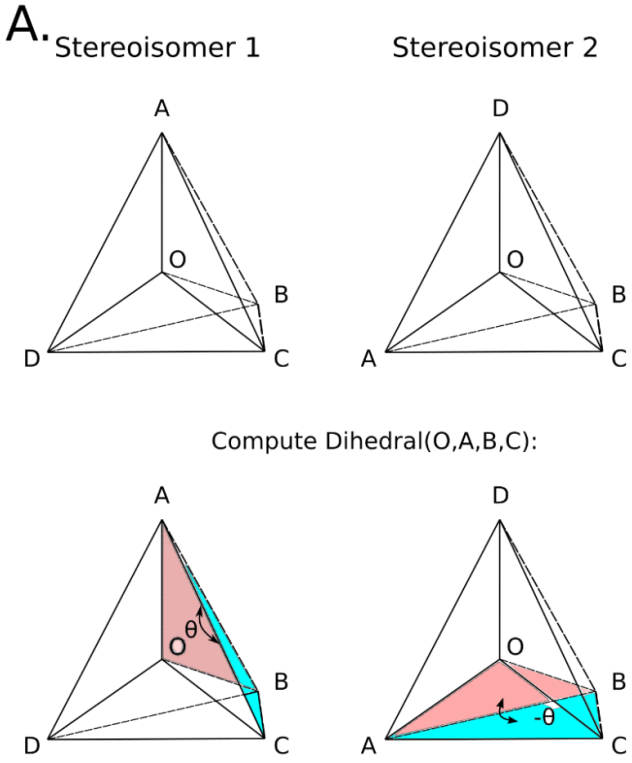


Figure 2.2: Depiction of chirality input angles. **A)** Chirality is encoded in the network by computing a set of dihedral angles of planes where the first plane starts with the chiral center. First row: 3D depiction of a chiral center with tetrahedral geometry with two different chiralities. Second row: Depiction of which angles are computed for each center. In practice, we compute the angles for all unique pairs of planes in the center that are explicitly modeled (hydrogens are implicit), measure their error from the ideal tetrahedron in the unit sphere, and pass the gradients of the error in predicted angles with respect to the predicted coordinates into the subsequent blocks as vector input features in the SE(3)-Transformer which breaks the symmetry over reflections present in the rest of the network and allows the network to iteratively refine predictions to match ideal tetrahedral geometry.

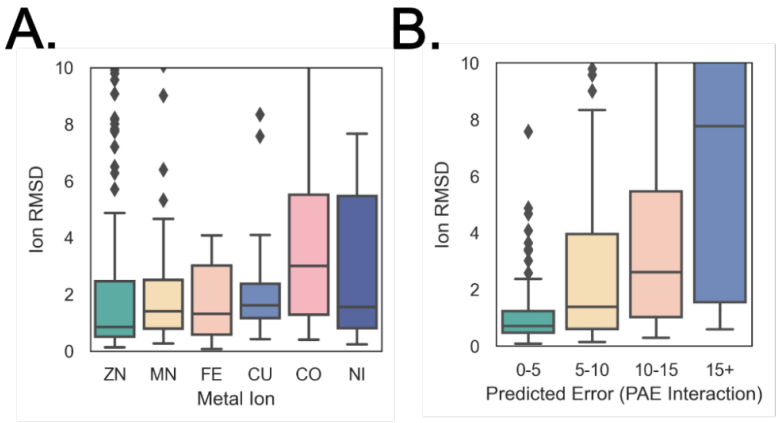


Figure 2.3: Performance on metal ions. **A)** Prediction accuracy on recent PDBs across 6 transition metals. **B)** Correlation between predicted error and Ion RMSD.

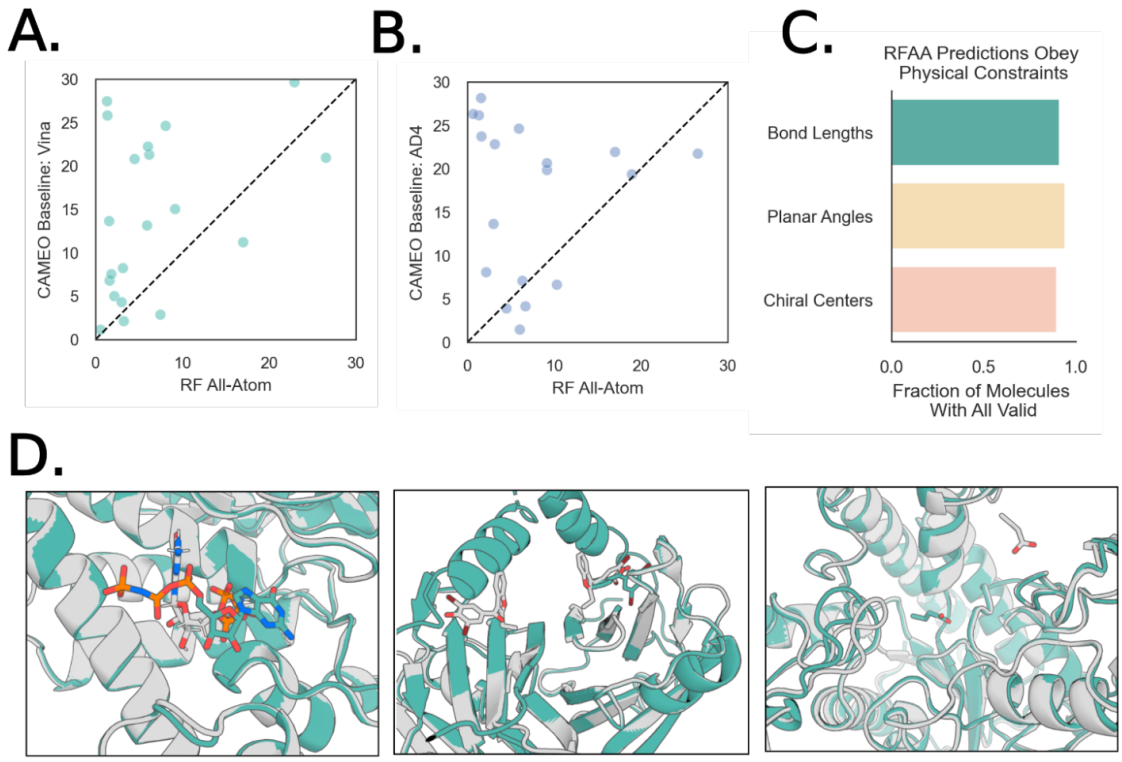


Figure 2.4: Additional results from the CAMEO ligand-docking challenge. **A)** Pointwise comparison of targets predicted both by RFAA and the CAMEO's baseline "Vina" server. RFAA predicts lower RMSD docks for a substantial number of ligand-protein complexes. **B)** The same comparison against CAMEO's "AD4" server. Again, RFAA outperforms the baseline server on the majority of the targets. **C)** RFAA preserves important structural properties of ligands in its predicted poses, such as accurate bond lengths between bonded atoms, planarity of aromatic rings and direction of chiral centers. **D)** Native structures are shown in gray and predicted structures are shown in teal. Some examples of high RMSD poses predicted by RFAA in the CAMEO challenge. From left to right: 1) The model predicts the correct global dock but orients the model incorrectly within the pocket (PDB ID: 7xql). 2) The model predicts an unresolved region as forming a pocket that interferes with the crystal dock of the ligand (PDB ID: 8ii2). 3) The model fails to predict the correct binding pocket of a small ligand, preferring to bury it deep into the protein (PDB ID: 8hwp).

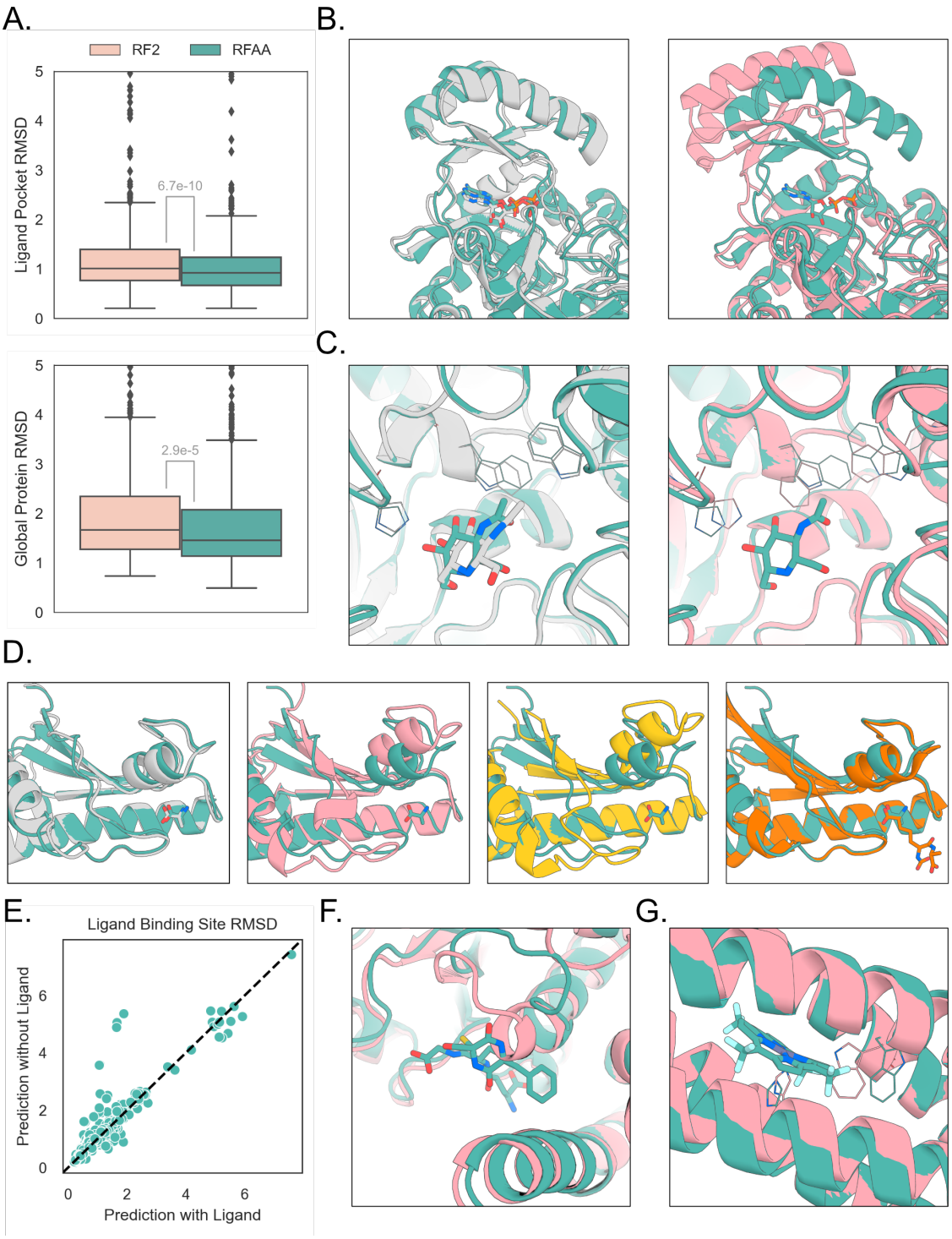


Figure 2.5: Comparisons to RoseTTAFold2 on predicting protein structures of ligand binding proteins. **A)** Both RFAA and RF2 were used to make predictions on a held-out set of ligand-binding proteins in the PDB (filtered by PAE < 10 and pLDDT > 0.8, respectively). RFAA makes better predictions than RF2, globally and on the pocket residues that bind to the ligand in the crystal structure (p-value < 0.05 paired t-test). **B)** Ligand-aware protein folding allows more confident positioning of relative protein domains on a flexible domain known to have a ligand-

induced conformational shift. The bound crystal structure is in gray, the RFAA prediction is green, and the RF2 prediction is pink (PDB ID: 7kct - closest seq. ID to training 82% for both models). The inter-domain PAE (residues 138-201) from the RF2 model is 9.63 vs. 6.26 for the RFAA model. **C)** Ligand-aware protein folding enables more accurate side-chain predictions in a binding pocket (PDB ID: 7mfl, closest seq. ID to training 29%). **D)** Structure prediction for the PDB entry 7rjj, along with the closest sequence match in the model's training set (yellow, PDB ID 5m38, seq ID 73%) and the sequence match in the model's training set with the most similar bound ligand (orange, PDB ID 5u1h, seq ID 40%). RFAA can use information from less similar proteins seen in training to make accurate predictions with small molecule context. **E)** Binding site RMSD of predictions of protein chains with and without ligands relative to the holo state crystal structure of the protein. The pink points are depicted in panels F and G. **F)** Ligand context helps the RFAA model to terminate a helix earlier in order to form a binding site for the ligand (PDB entry 7nc6, closest seq. ID to training 32%). **G)** Prediction of a designed porphyrin-binding protein (PDB entry 7jh6, closest seq. ID to training 43%). The pocket structure was designed based on 5tgy, which is also not present in the RFAA training set. The model adjusts bulky side-chains in the pocket to allow space for the bound porphyrin.

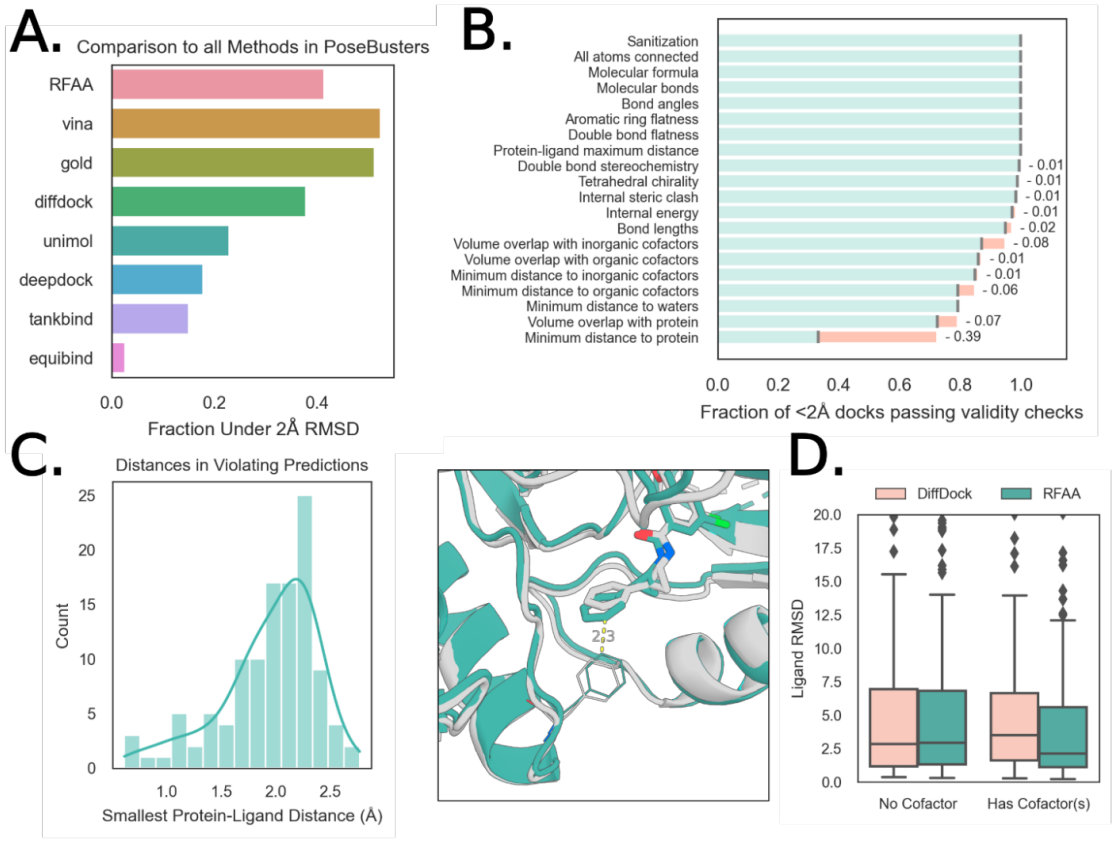


Figure 2.6: Analysis of predictions of recent PDB protein-small molecule complexes. **A)** Comparison to other methods on PoseBusters benchmark set, which attempts to assess the validity of predicted structures from deep learning networks. Each model is used in its "training regime," which means that Vina and Gold receive the bound crystal structure and a bounding box for the pocket. RFAA only receives the protein sequence and basic atomic graph information for the small molecule. **B)** Waterfall plot showing results from running the PoseBusters filters on predicted structures from RFAA. Most structures are physically plausible, with most violations occurring in the "minimum distance to protein" metric. **C)** Distribution of shortest distance to protein in violating predictions. Most "violating" distances are between 2.0 and 2.5 Å. The second panel shows an example of a violating distance at 2.3Å. **D)** Comparison of RFAA and DiffDock on cases with and without cofactors. In cases with cofactors, RFAA outperforms because of its ability to simultaneously model multiple molecules.

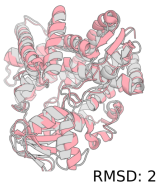
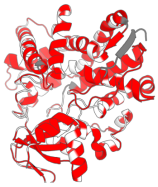
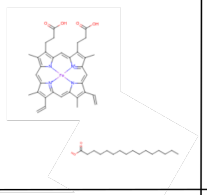
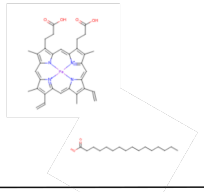
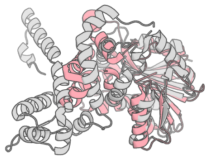
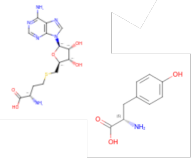
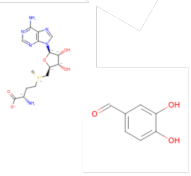
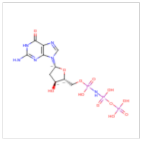
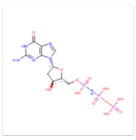
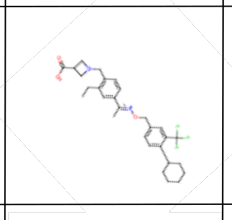
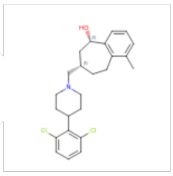
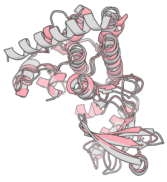
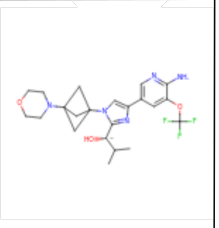
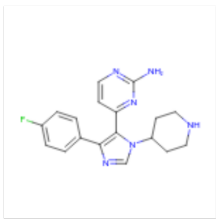
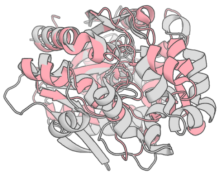
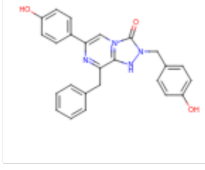
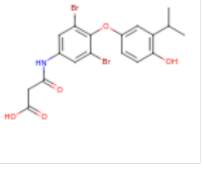
Main Text Figure	Closest protein (Seq ID%) in Train Superposition	Closest protein in Train colored by mutations	Ligand(s)	Closest Ligand in Train with Similar Protein
Figure 2D (left)	 RMSD: 2.5A			
Figure 2D (middle)	 RMSD: 4.4A			
Figure 2D (right)	 RMSD: 0.5A		 DNA Mg	 DNA Mg
Figure 2H (left)	 RMSD: 3.8A			
Figure 2H (middle)	 RMSD: 2.7A			
Figure 2H (right)	 RMSD: 4.4A			

Figure 2.7: Column 1: Structural superimposition to closest training structure (RMSD computed by PyMOL). Gray: structure shown in example, Pink: closest structure (by SeqID%) in training set. Column 2: Mutation profile of closest training structure compared to example in Figure 2. Red: mutated residue, White: no mutation, Gray: did not align (in PyMOL). Column 3: Ligands shown in figure example. Column 4: Closest ligand (by Tanimoto similarity) to all ligands that bind any BLAST hit in the training set.

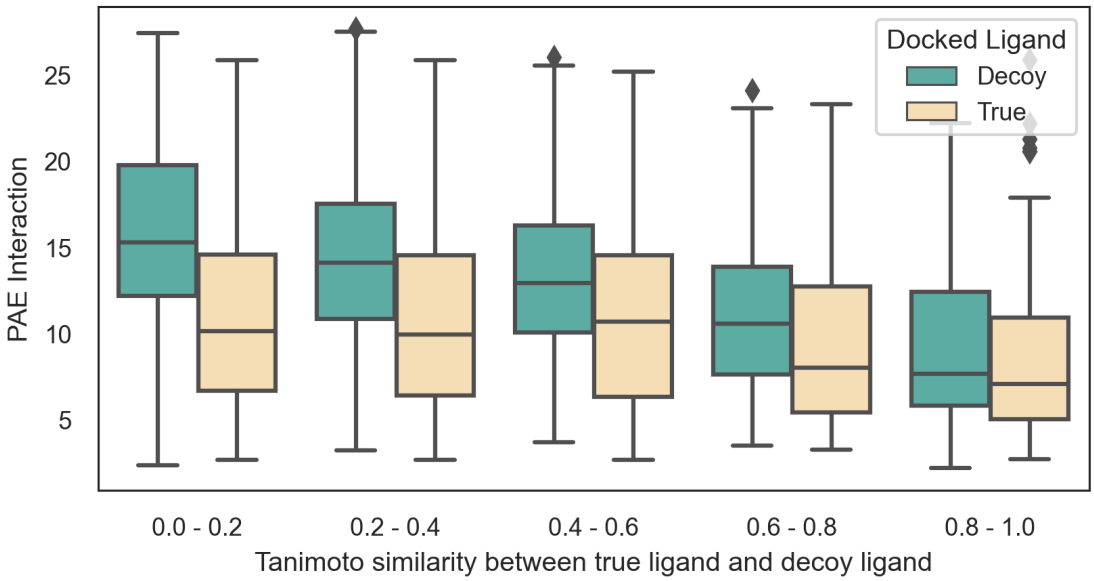


Figure 2.8: Cross-docking "decoy" ligands on the Posebusters test set. 6 decoy ligands - the ligand with the highest Tanimoto similarity to the "true" ligand plus 5 chosen uniformly random from the 42 off-target ligands in Posebusters - were predicted in complex with each protein target in Posebusters. The model's predicted error (PAE Interaction) of the decoy ligand dock increases when the decoy ligand is not structurally similar to the true ligand. In particular, 75.1% of decoy docks have a higher predicted error than the true dock.

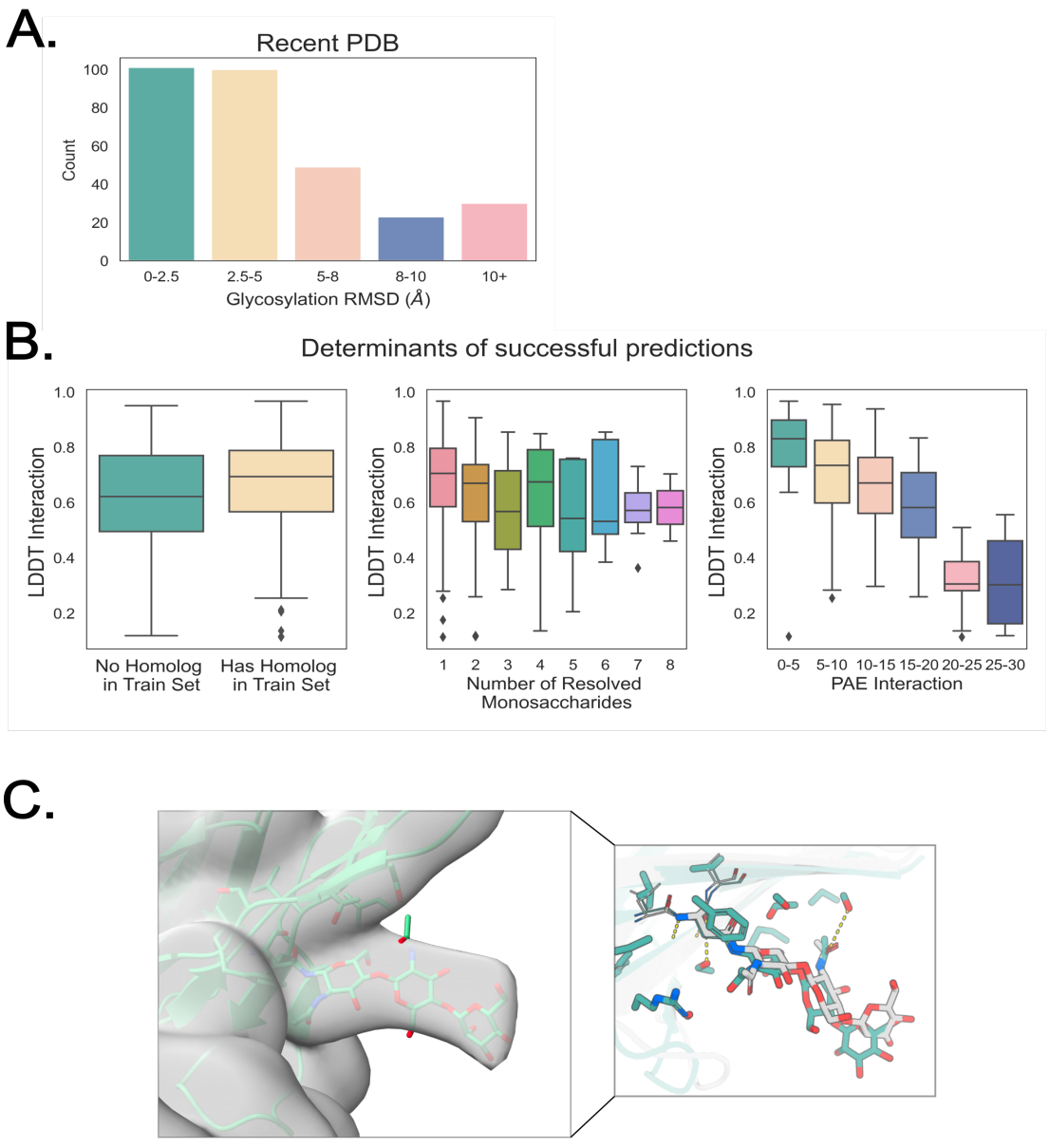


Figure 2.9: Analysis of predictions of glycosylated proteins. **A)** Histogram of RMSDs of predictions made on recently solved glycoproteins. **B)** Determinants of successful predictions from left to right. LDDT Interaction is computed by measuring the all-atom LDDT between protein residues and the residue and modification atoms. From left to right: Having a homolog in the training set (>30% sequence similarity) does not seem to be a large indicator of successful predictions. While predictions of shorter glycans are more accurate, longer glycans can still be successfully modeled. The model's error prediction accurately identifies high-accuracy predictions. **C)** (Left) Predicted glycoprotein (PDB ID: 7u7n), the IL-27 quaternary signaling complex fits well into Cryo-EM density. (right) Network accurately predicts hydrogen bonds between sidechains and glycan (Native structure in gray, predicted structure in teal).

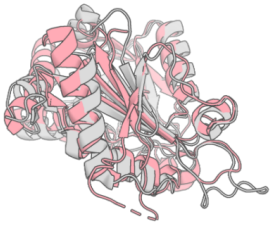
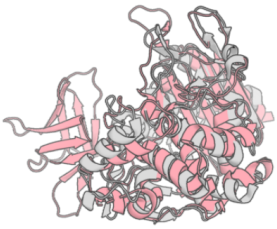
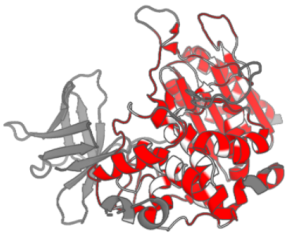
Main Text Figure	Closest protein (Seq ID%) in Train Superposition	Closest protein in Train colored by mutations
Figure 3E (top)	 RMSD: 2.7A	
Figure 3E (bottom)	 RMSD: 3.5A	
Figure 3F (left)	No BLAST hits	No BLAST hits
Figure 3F (right)	 RMSD: 9.9A	

Figure 2.10: Column 1: Structural superimposition to closest training structure (RMSD computed by PyMOL). Gray: structure shown in example, Pink: closest structure (by SeqID%) in training set. Column 2: Mutation profile of closest training structure compared to example in Figure 2. Red: mutated residue, White: no mutation, Gray: did not align (in PyMOL).

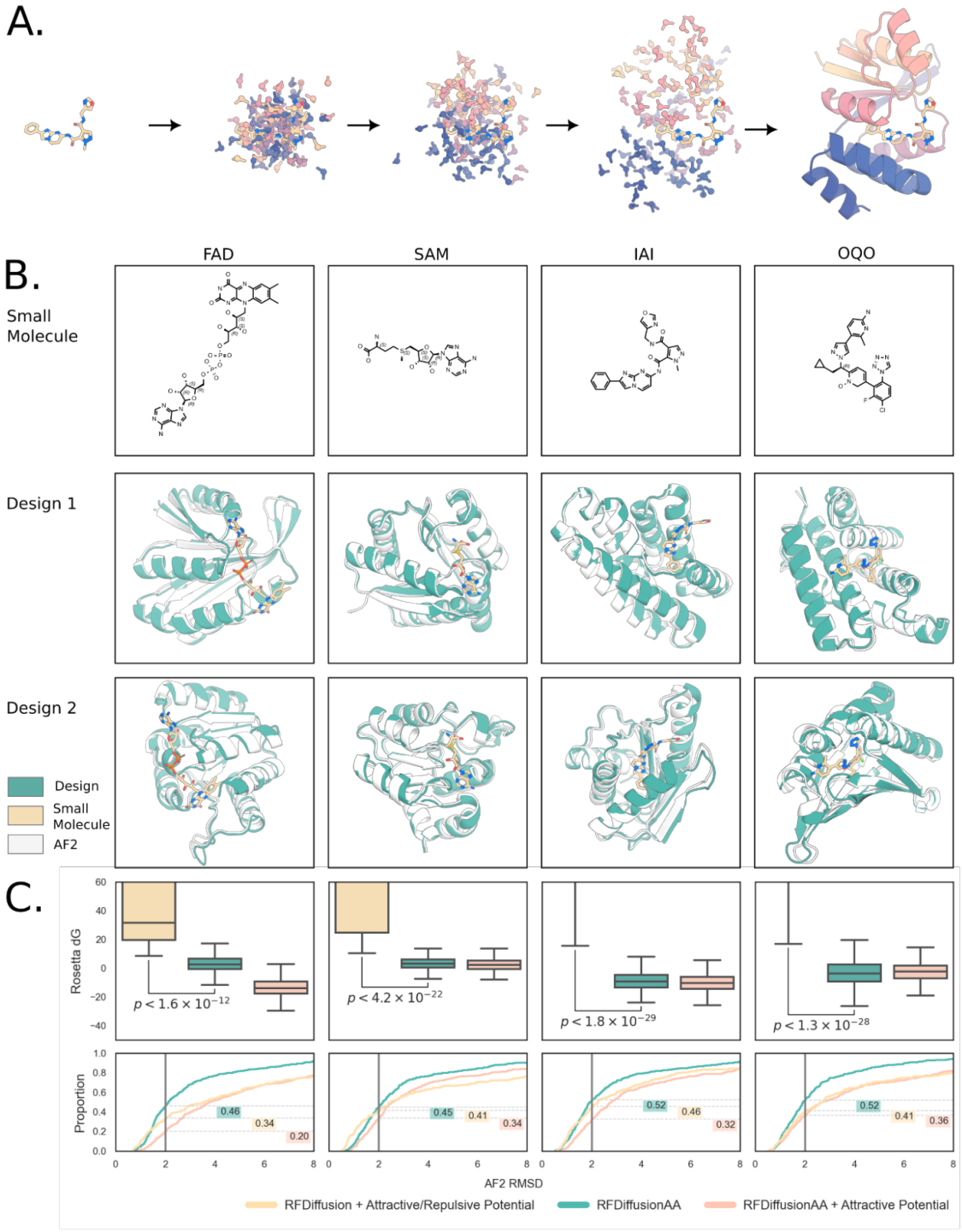


Figure 2.11: Small molecule binding protein design with RFdiffusion All-Atom **A)** Schematic depicting the denoising of a residue gas into a small molecule binder. **B)** Binder design models for four representative small molecules (top row) are nearly identical to the AF2 pose prediction from single sequence alone. Two models are shown (second and third rows) to illustrate design diversity (for more extensive evaluation of diversity, see Figure 2.12). **C)** Comparison of protein-only

RFdiffusion with the substrate modeled implicitly using an attractive/repulsive potential against RFdiffusionAA with and without attractive potential. Top: GALigandDock evaluated binding δG (minimum of eight LigandMPNN sequences). In all cases, RFdiffusionAA produces protein-ligand interfaces with a lower computed δG than RFdiffusion. Bottom: proportion of designs with RMSD to AF2 prediction less than the value specified by the x-axis (minimum of eight LigandMPNN sequences). The fraction of designs with less than 2Å RMSD to AF2 are highlighted; in all cases, RFdiffusionAA produces more self-consistent designs than RFdiffusion. Two of the small molecules are quite distinct from any of the molecules in the training set (Tanimoto similarity ≤ 0.5; IAI: 0.50, OQO: 0.46); the design metrics for the binders to these models are similar to those of previously seen ligands, suggesting that the model has considerable ability to generalize not only in protein topology and structure but also in interactions with non-protein targets.

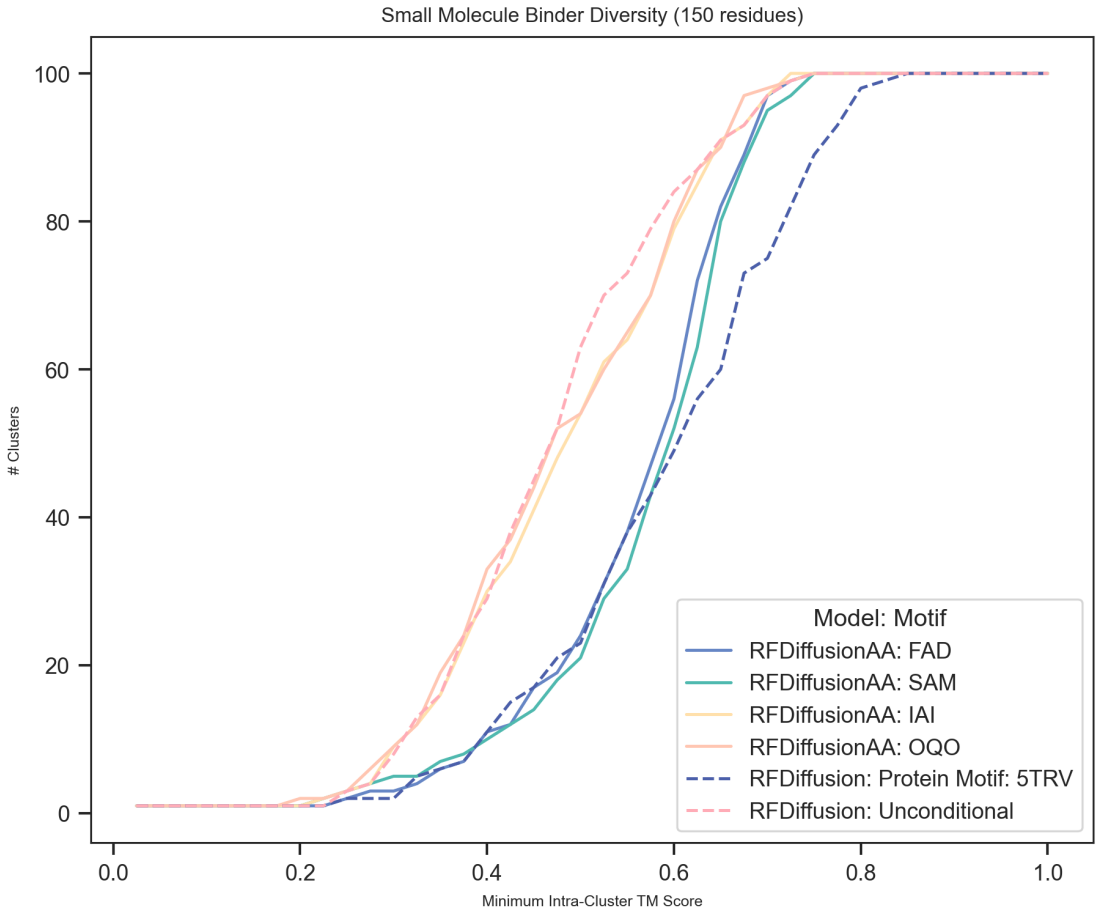


Figure 2.12: Diversity of small molecule binders generated by RFdiffusionAA. To assess diversity among the generated binders for a given task we perform an all-by-all TMAlign of 100 unfiltered designs. We then perform agglomerative clustering at TM-score thresholds from 0 to 1, such that every design within a cluster has a TM-score to every other design in that cluster less than that threshold and report the number of clusters. Left shifted curves correspond to more clusters at less stringent clustering criteria and thus more diverse designs. We observe that for ligands that appear frequently in the training set [FAD, SAM], the binders generated are less diverse than those designed against ligands with low similarity to any ligand in the training set [IAI, OQO] as the network has come to recognize some of the canonical binding modes of common ligands. To contextualize the magnitude of this difference in diversity we show the same analysis using RFdiffusion when generating unconditional samples vs. RFdiffusion generating motif-conditional samples. RFAA designs have comparable diversity to conditional samples from RFdiffusion.

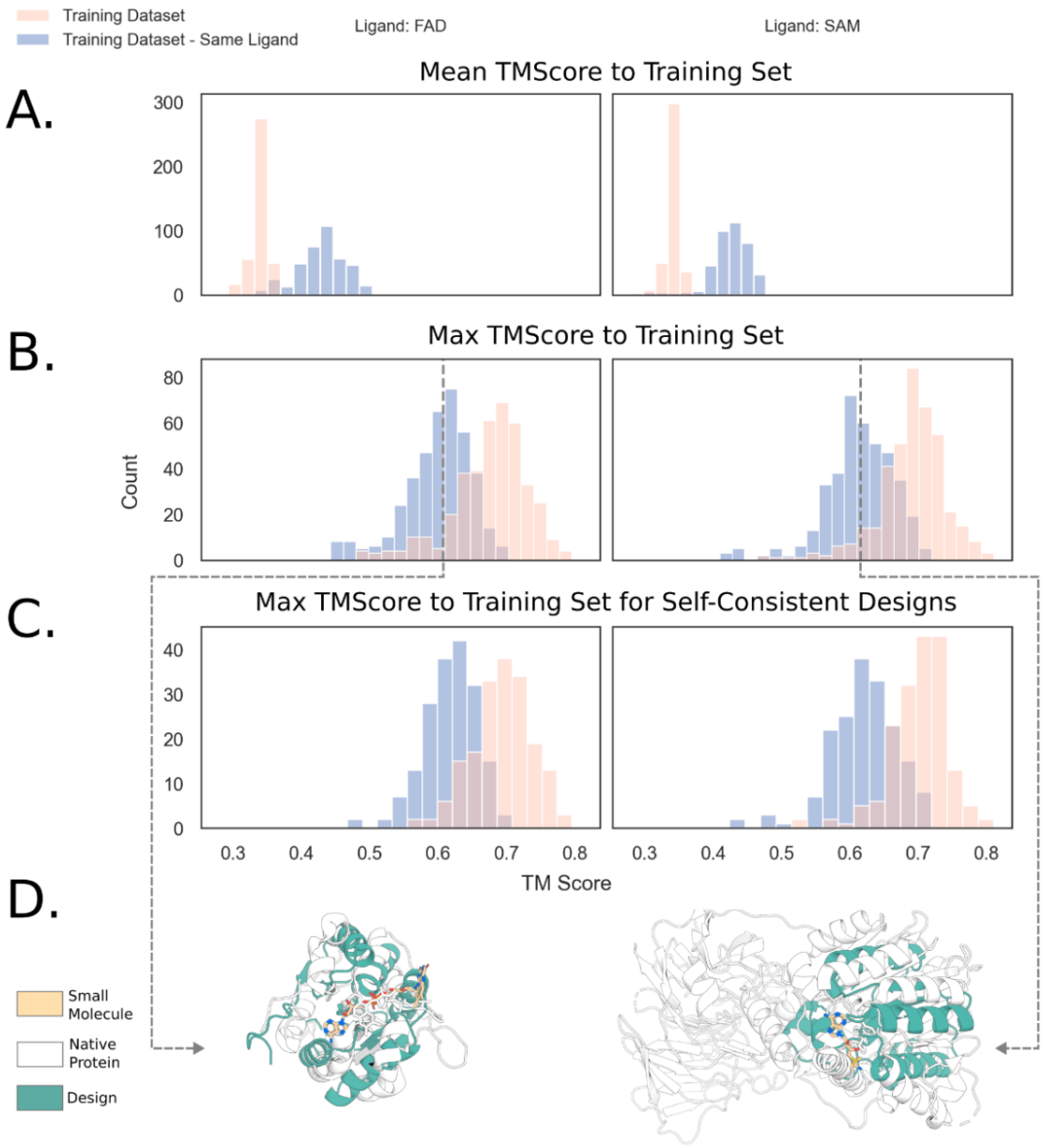


Figure 2.13: A) Mean TM-score to all hits in the training dataset for each of 400 designs. Designs against an already-seen ligand have a higher mean TM-score on averages to proteins in the training dataset that possess the ligand than to the training dataset as a whole. **B)** Maximum TM-score to the training dataset for each of the 400 designs. The most similar protein to a design in the entire training dataset is substantially more similar than any protein in the training dataset which possesses the ligand, i.e. designs made against an already-seen ligand are not memorized examples from the training dataset for that ligand. **C)** Designs filtered to only those that are self consistent (AF2 RMSD < 2Å), to show that the novelty demonstrated in (B) is not owed to backbones that would not fold. **D)** Teal: The design with median TM-score to the training set with the same ligand bound, White: Closest PDB structure with the same ligand bound.

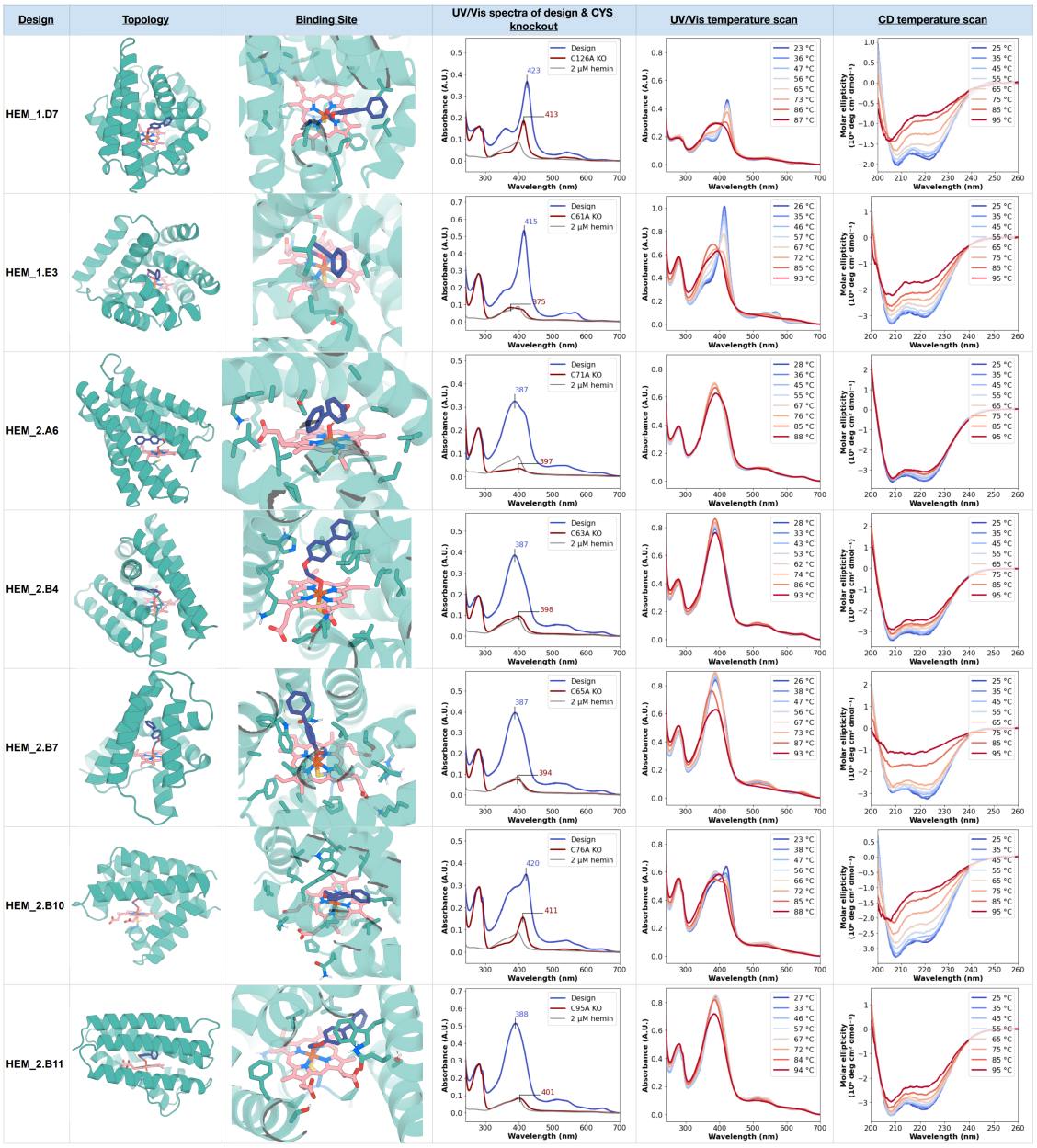


Figure 2.14: Characterization of heme-binding proteins obtained with RFdiffusionAA, HEM_1.D7 to HEM_2.B11. The 4th column shows the UV/Vis spectra of the purified heme-loaded protein at approximately 10 μM (blue), and the putative axial Cys/Ala mutant at 10 μM, mixed with 2 μM hemin (red trace), along with free hemin at 2 μM (gray). The 5th column shows changes in the UV/Vis spectra, upon heating the protein sample to above 86 °C. The last column shows the CD spectra at increasing temperatures up to 95 °C.

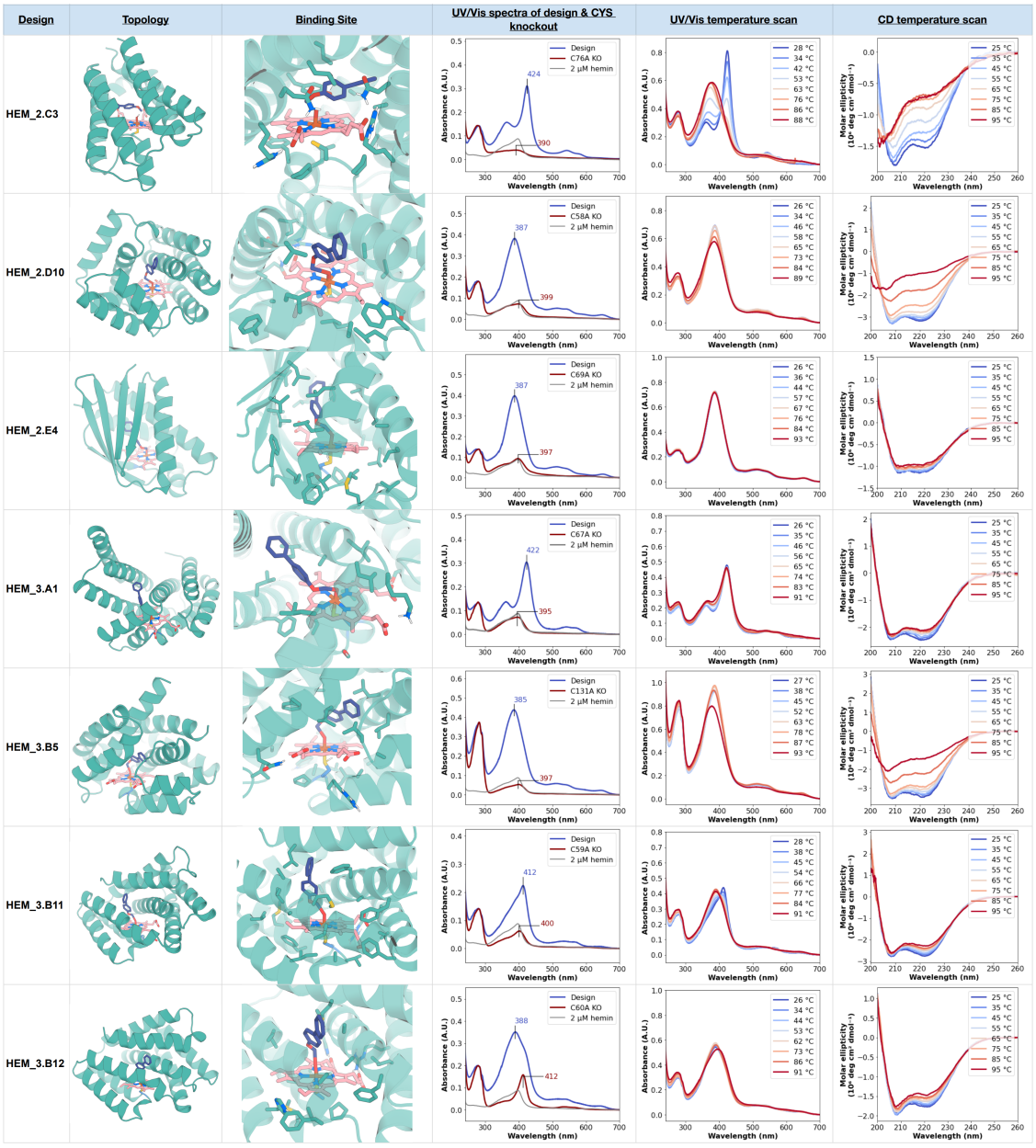


Figure 2.15: Characterization of heme-binding proteins obtained with RFdiffusionAA, HEM_2.C3 to HEM_3.B12. The 4th column shows the UV/Vis spectra of the purified heme-loaded protein at approximately 10 μM (blue), and the putative axial Cys/Ala mutant at 10 μM, mixed with 2 μM hemin (red trace), along with free hemin at 2 μM (gray). The 5th column shows changes in the UV/Vis spectra, upon heating the protein sample to above 86 °C. The last column shows the CD spectra at increasing temperatures up to 95 °C.

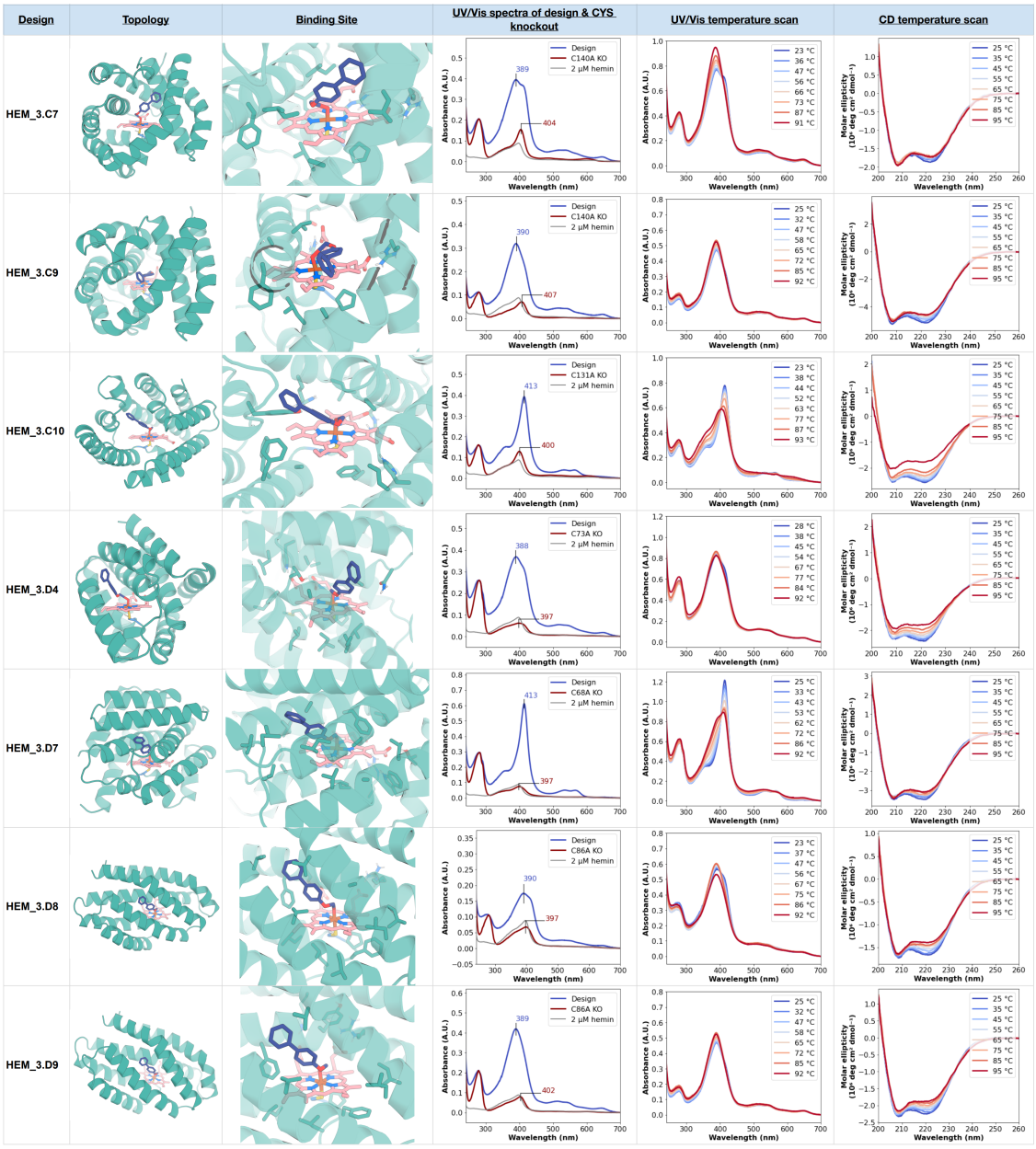


Figure 2.16: Characterization of heme-binding proteins obtained with RFdiffusionAA, HEM_3.C7 to HEM_3.D9. The 4th column shows the UV/Vis spectra of the purified heme-loaded protein at approximately 10 μM (blue), and the putative axial Cys/Ala mutant at 10 μM, mixed with 2 μM hemin (red trace), along with free hemin at 2 μM (gray). The 5th column shows changes in the UV/Vis spectra, upon heating the protein sample to above 86 °C. The last column shows the CD spectra at increasing temperatures up to 95 °C.

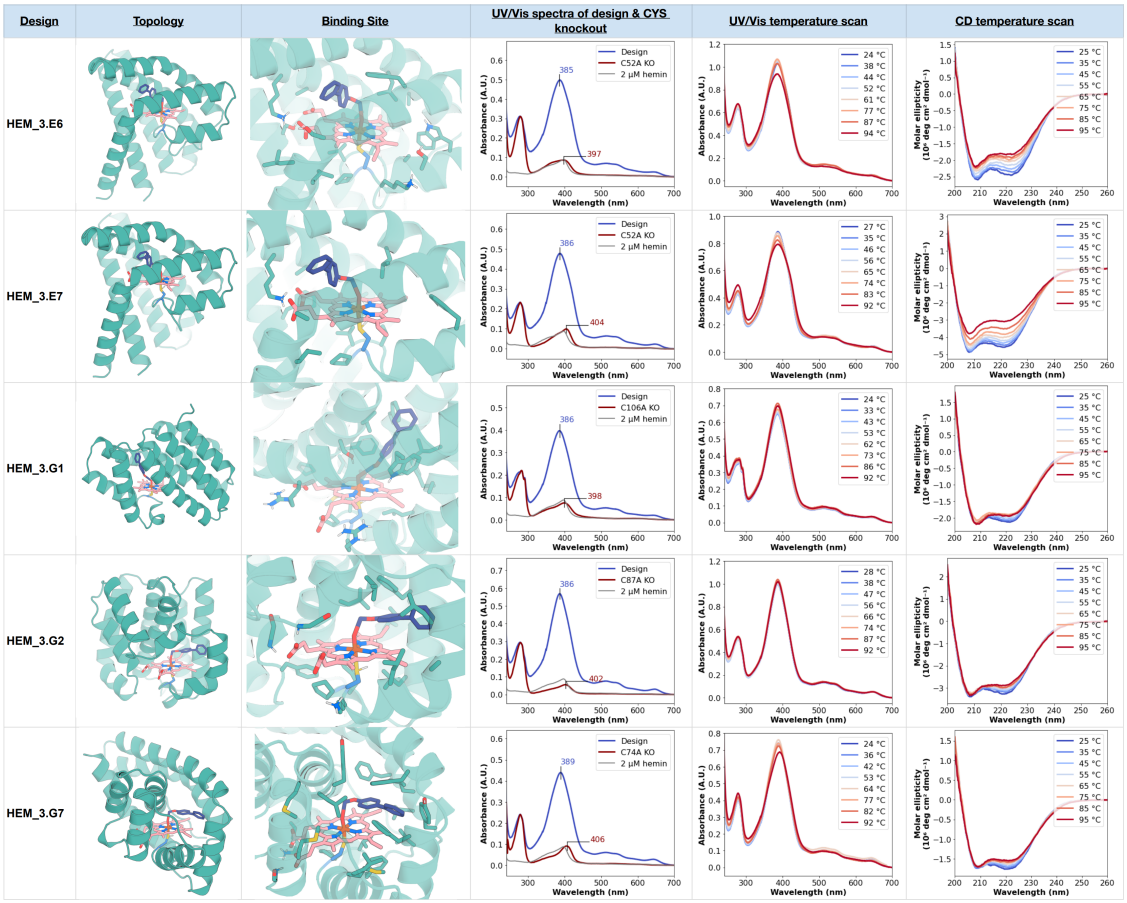


Figure 2.17: Characterization of heme-binding proteins obtained with RFdiffusionAA, HEM_3.E6 to HEM_3.G7. The 4th column shows the UV/Vis spectra of the purified heme-loaded protein at approximately 10 μM (blue), and the putative axial Cys/Ala mutant at 10 μM, mixed with 2 μM hemin (red trace), along with free hemin at 2 μM (gray). The 5th column shows changes in the UV/Vis spectra, upon heating the protein sample to above 86 °C. The last column shows the CD spectra at increasing temperatures up to 95 °C.

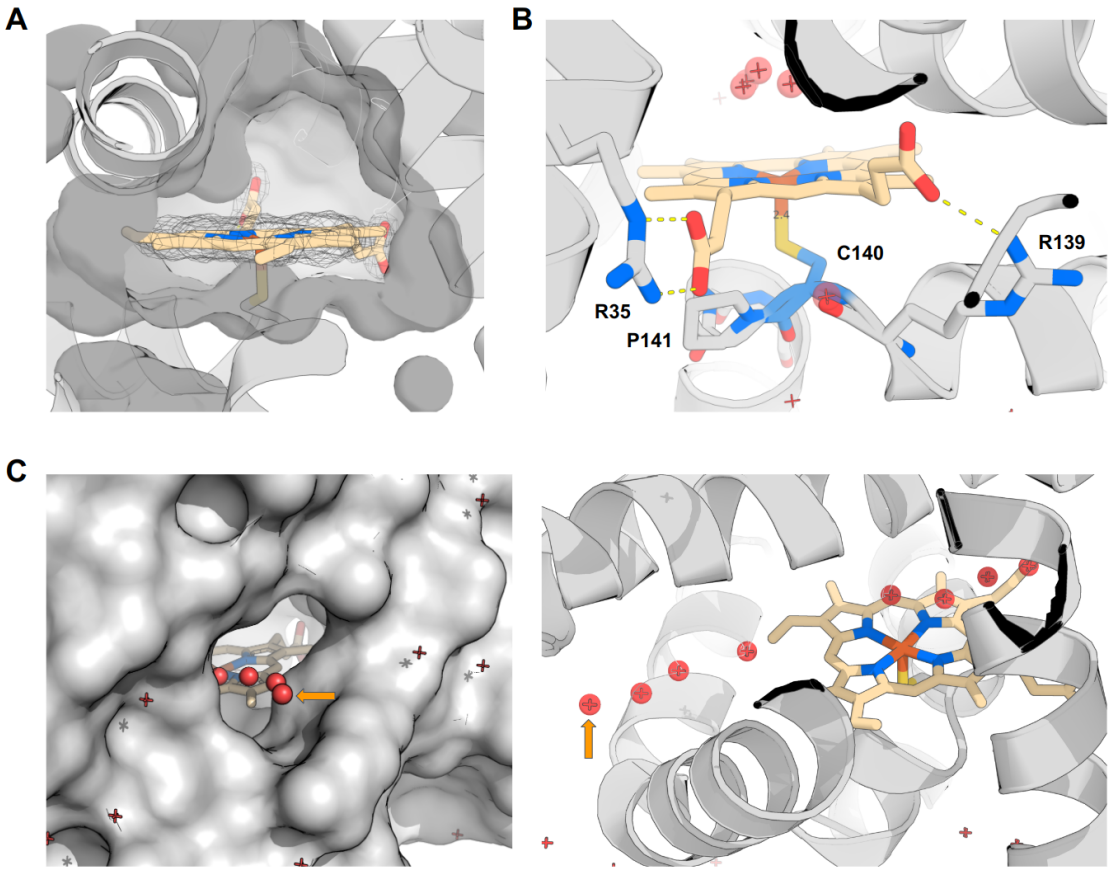


Figure 2.18: Crystal structure of **HEM_3.C9** (PDB id 8vc8). **A)** Electron density corresponding to heme (2Fo – Fc omit map contoured at 1σ). **B)** Detailed view of the heme-binding motif. Heme is ligated by Cys-140 with Fe-S distance of 2.4 Å. Arg-139 extends out to form a hydrogen bond with one of the heme propionates. Cys-140 and Pro-141 form the N-terminus of a helix, akin to the binding motif in native unspecific peroxygenases. Cys-140 sulfur atom is further supported by a hydrogen bond with the backbone amide NH of Leu-143 (r(N-S) = 3.3 Å). **C)** A water-filled channel connecting the heme binding site with the surface of the protein. Orange arrow denotes equivalent water molecules (red spheres) in the left and right panels.

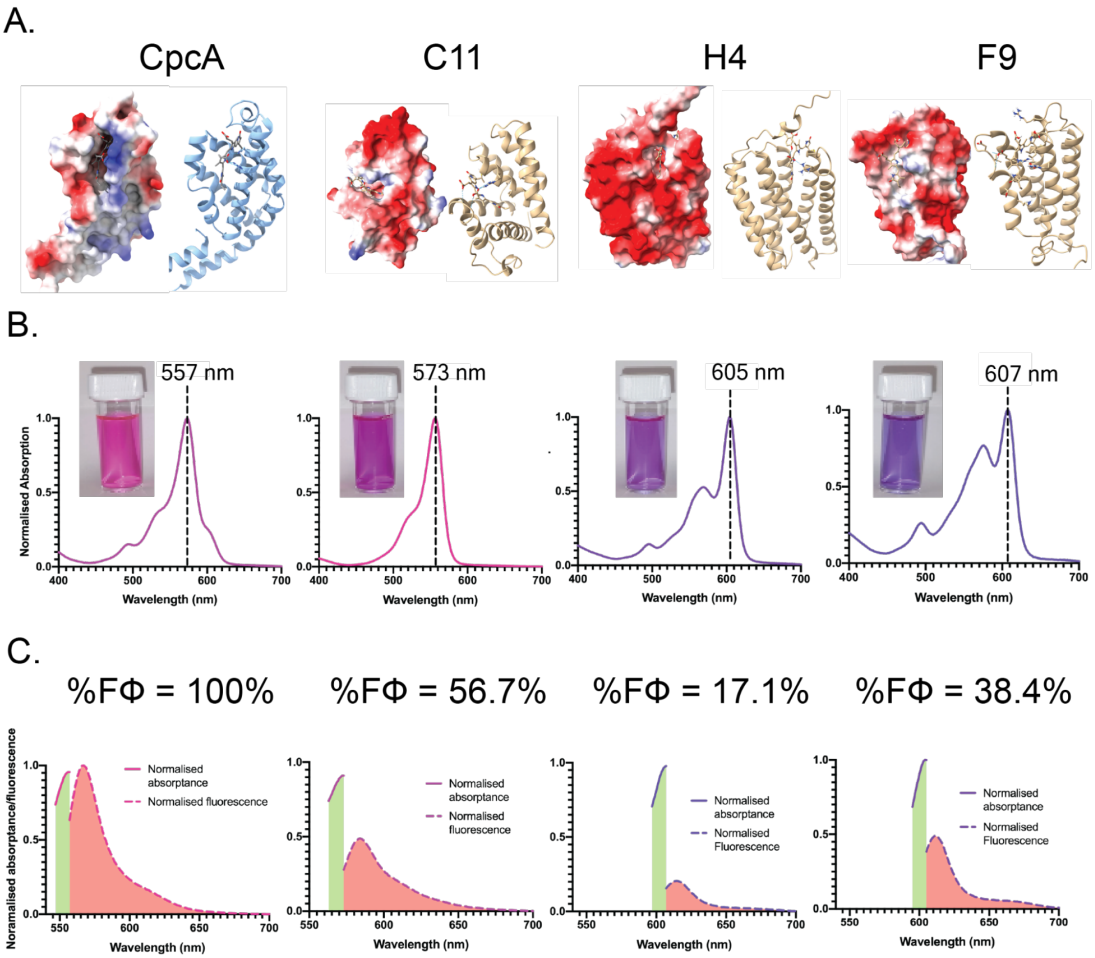


Figure 2.19: Designed biliproteins. **A)** Molecular topologies and electrostatic space-filling models; molecular topologies of the positive control (CpcA) and three RFdiffusionAA designs (C11, F9, and H4) and their electrostatic space-filling models (blue = positive; red = negative). **B)** Normalised absorption spectra of CpcA, C11, F9, and H4 with images of the colored purified protein solutions. **C)** Normalised absorptance and relative emission profiles of Cpca and the designed proteins. These measurements were used to calculate the relative fluorescence yield of the designed proteins (C11, H4, and F9 - 57.6%, 17.1%, and 38.4%, respectively) with CpcA set as 100%. Green area under the graph expressed as absorptance and red area under the graph as fluorescence.

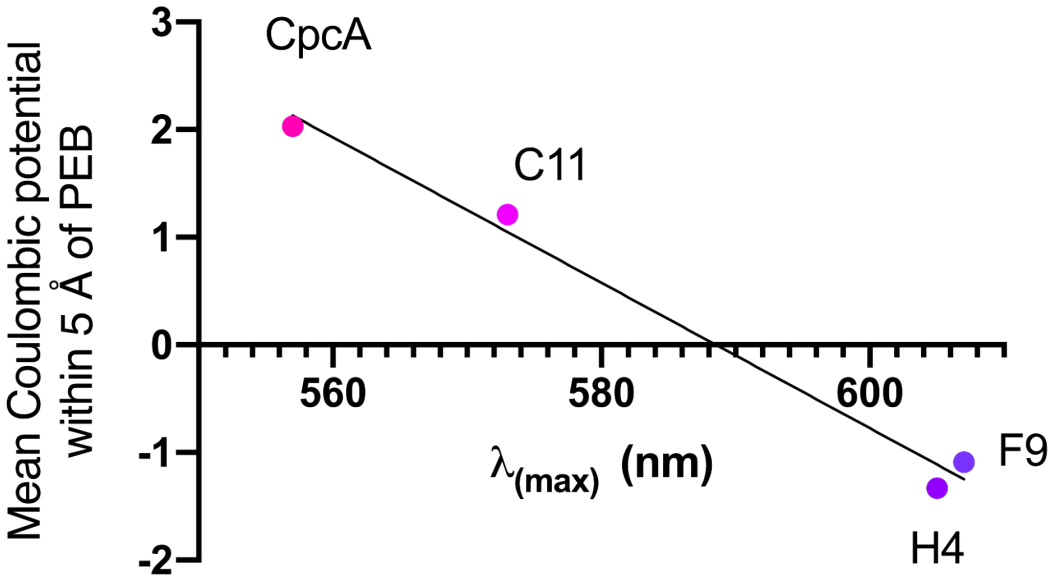


Figure 2.20: Correlation between coulombic charge surrounding bilin and the maximal absorption wavelength.

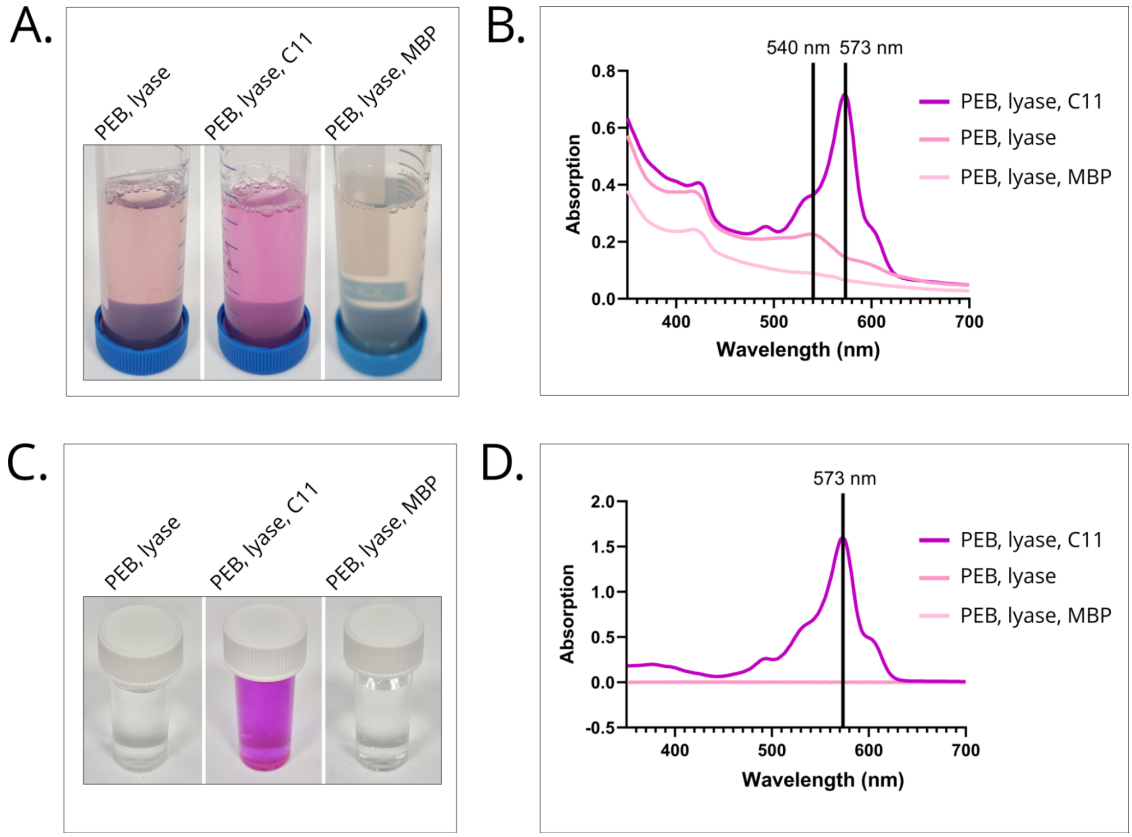


Figure 2.21: Comparison of PEB bilin absorption in the absence of a binding protein, with the C11 designed protein, and with a non-binding control protein. Three *E. coli* BL21(DE3) strains harbored a plasmid for synthesizing PEB and the CpcE/F bilin lyase. One strain contained just this plasmid, the second also contained a plasmid for the synthesis of C11 or maltose binding protein (MBP). Cells were grown in 500 ml of LB medium and gene expression was induced when culture reached OD600 0.6 using 1mM IPTG for 16-18 h at 16 °C. Cells were resuspended in lysis buffer (50 mM HEPES, 500 mM NaCl pH 7.6 with protease inhibitors, DNase I and lysozyme), sonicated and centrifuged to generate cell-free extracts (CFEs). **A)** Image of CFEs. **B)** Absorption spectra of theCFEs showing that PEB only acquires strong 573 nm absorption in the presence of C11. The weak 540 nm feature arises from PEB bound to the lyase. **C)** Image of eluates following fractionation of supernatants by immobilized metal chelate affinity chromatography. MBP was purified on an amylose column. **D)** Absorption spectra of the eluates. The featureless spectra for the PEB+lyase and PEB+lyase+MBP samples appear as overlapping flat lines. The presence of the highly absorbing band at 573 nm shows that C11 retains tightly bound PEB during purification.

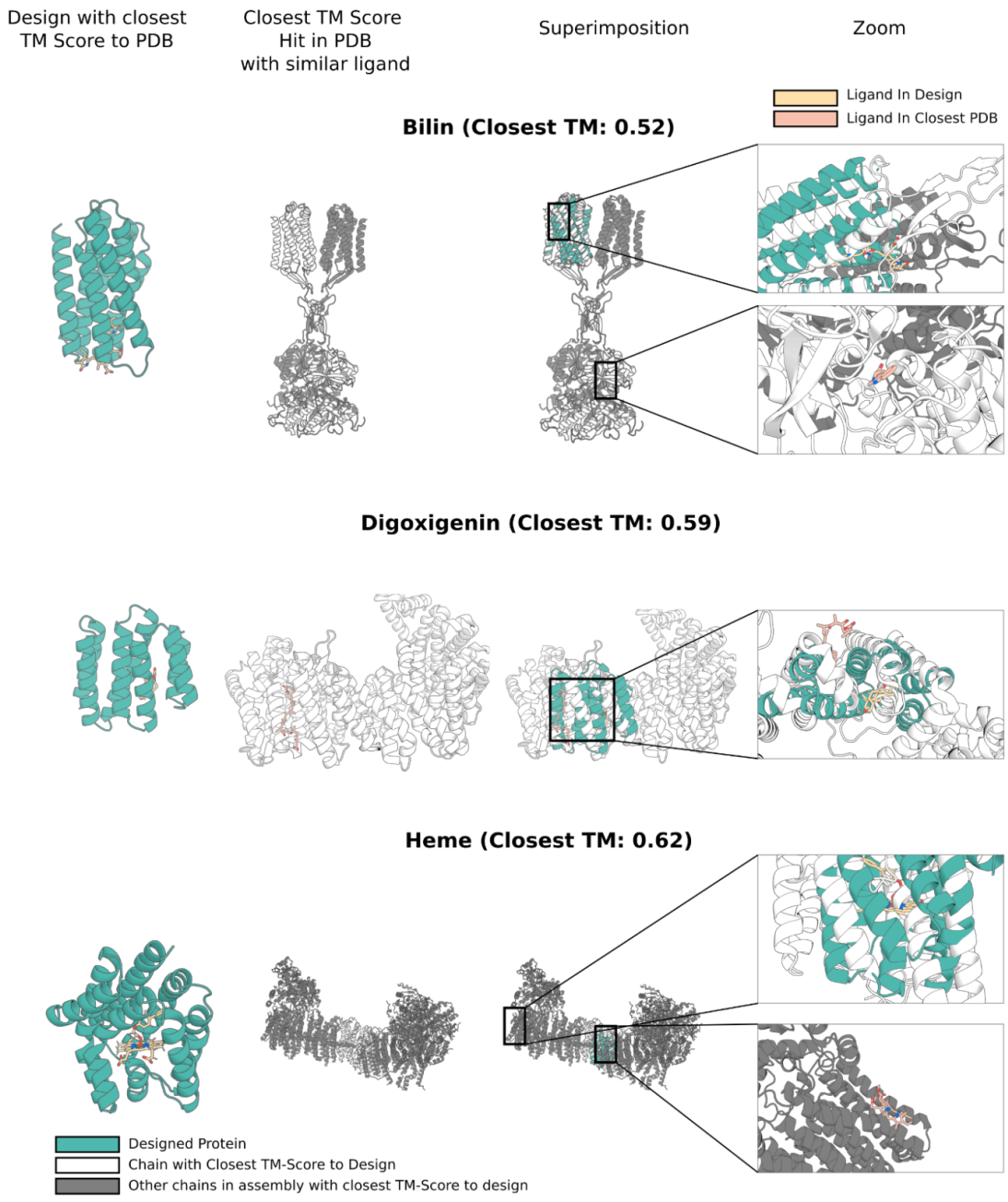


Figure 2.22: Novelty of experimentally characterized proteins. For each design case, we found the design with the highest TM score to a protein chain in the training set out of all experimentally successful designs. For all hits in the training set, we measure whether a similar ligand (Tanimoto > 0.5) is present in the PDB entry containing the TM align hit. For each design task, we show the designed protein, the entry with the closest TM score and bound to a similar ligand, the superimposition of the design into the closest TMalign hit and finally a zoom into the binding site of the designed interface (ligand in yellow) and interface with the similar ligand in the training set (ligand in pink). In all cases, the closest entry with a similar ligand in the training set by TM score is below 0.62 and the binding mode of the similar ligand is quite different compared to the binding mode for the designed protein.

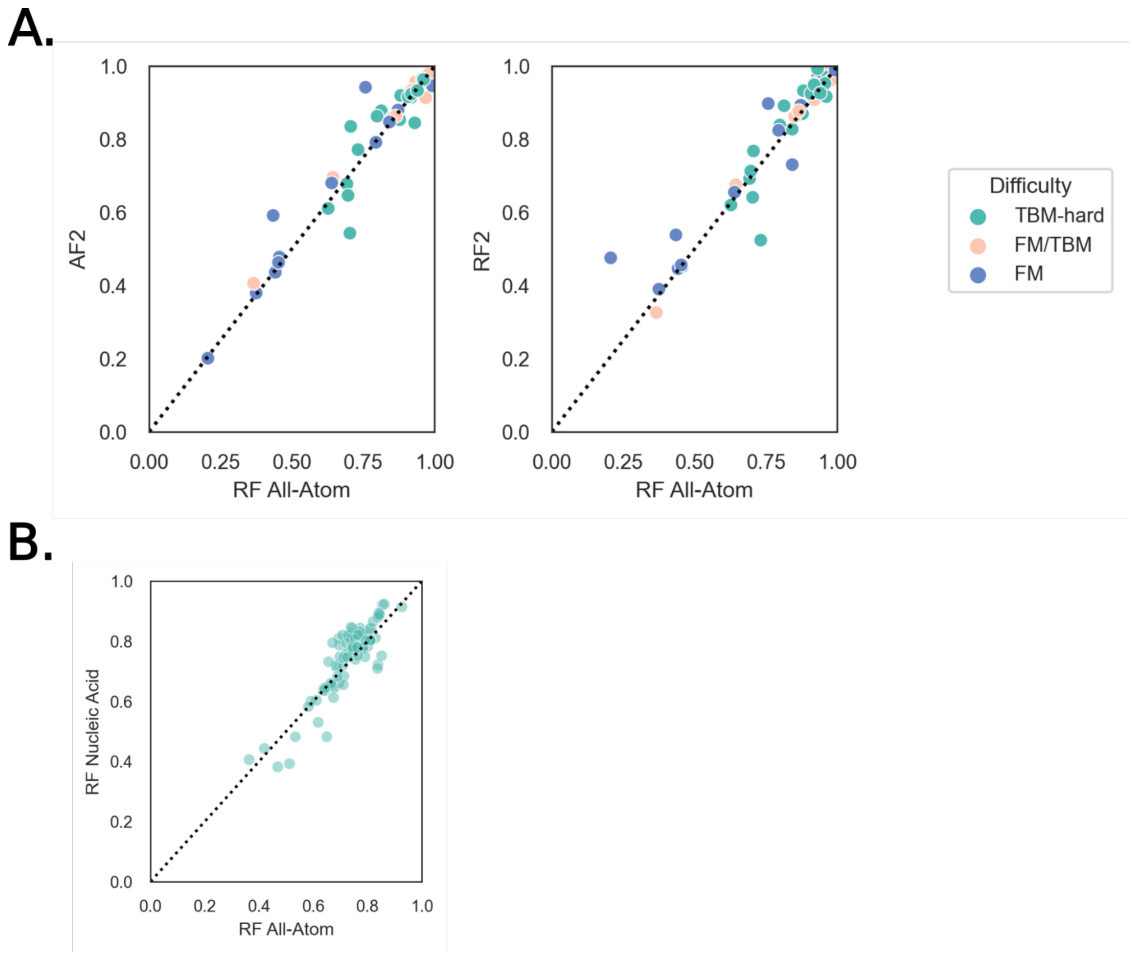


Figure 2.23: Comparisons to AlphaFold2 and RoseTTAFoldNA on protein structure prediction and protein-nucleic acid interface prediction. **A)** Comparison of RFAA to AF2 and RF2 on protein monomer structure prediction based on a set of 126 structures from CASP14 (TBM-hard, FM/TBM, and FM). Each prediction is scored on GDT to the native structure and colored by difficulty. **B)** Comparison of RFAA to RFNA on protein-nucleic acid complex prediction on a dataset of 89 recently solved structures that were not in the training set of either method.

2.10 Materials and Methods

2.10.1 Dataset Curation

The datasets used to train RosettaFold All-Atom can be broadly grouped into three categories: protein-only, nucleic acid, and small molecule datasets. In this subsection, we describe the curation of each dataset. During training, protein sequence clusters are sampled within each dataset (further discussion of how often each dataset is sampled in subsection 2.10.6). Multiple Sequence Alignments (MSAs) and templates were generated as in RoseTTAFold2 (RF2) [10]. For MSAs, hhblits [108] was ran at successive e-value cutoffs until 10000 unique sequences with >50% coverage were found. For templates, HHsearch [115] was ran to find a maximum of 500 templates with probability >5%.

Protein-Only Datasets

Datasets with protein-only examples come from both the Protein Data Bank (PDB) [17] and the AlphaFold2 (AF2) [72] distillation set.

PDB Monomers and Protein Complexes Similar to RF2 [10], we train on protein monomer and protein complexes structures deposited into the PDB before April 30, 2020 with resolution below 4.5Å. For each chain, we find all contacting chains in all bioassemblies and featurize pairs of homomeric and heteromeric proteins by cropping around the interface. In cases where heteromeric complexes are from the same organism, we provide paired MSAs.

AlphaFold 2 Distillation Data We train RFAA on a set of UniRef50 structures predicted by AlphaFold2 in [66]. We follow RF2 and augment our training data with structures with AF2 mean pLDDT >70. We apply backbone losses for all structures but only apply sidechain losses for residues with AF2 pLDDT >90. The structures were predicted with only MSAs and no templates and we do not use templates for these examples in training.

Nucleic Acid Datasets

We follow RoseTTAFold nucleic acid (RF-NA) [12] train on protein-nucleic acid complexes and RNA structures. The training setup is identical to the training set for RF-NA.

Small Molecule Datasets

This subsection deals with data that involves non-polymer small molecules (or non-linear polymers such as sugars), generally bound to a protein. Each item in the various small molecule datasets is represented by a selected subset of chains from a PDB entry. We build the dataset by iterating through each entry in the PDB with a bound, non-polymer chain. For each non-polymer chain, we first discard that entry if it is considered "non-biological" (see subsection 2.10.1). Otherwise, we treat that chain as a *query ligand* of interest and compute a list of every chain in that entry that contacts it (see subsection 2.10.1). We maintain a running list of transformations operate on each chain's coordinates to place it into a global reference frame: this is necessary for symmetric assemblies in the PDB where a single chain may represent multiple copies of identical molecules in different, symmetric positions. In all cases, hydrogens are not modeled. Finally, we compute MSAs and templates for each protein chain contacting each query ligand in the same manner as in subsection 2.10.1.

This definition of dataset item implies that a single PDB entry can comprise of multiple items in the dataset corresponding to different query ligands. For example, the PDB entry '5nag' corresponds to two entries in our dataset: one for each bound ligand (FAD and 8R5). At training time, we crop each entry around the query ligand (further description in subsection 2.10.3), resulting in different contexts for the two query ligands.

We also note here that we only include in our small molecule datasets those PDB entries that are present in either the PDBBind [128] or BioLip [141] datasets.

Ligand Filtering We find that many of the non-polymer entities in PDB entries are *non-biological*: that is, they represent solvents or crystallization additives rather than binding partners found in a biological context. We curated a list of 3-letter component identifiers

Table 2.1: Nonbiological Molecules

NUC, ZN, CA, MG, III, MN, FE, CU,
SF4, FE2, CO, FES, GOL, NA,
CL, K, CU1, GOL, XE, NO2, EDO,
NI, BR, CD, O, CS, NO, TL, HG, UNL, KR,
SR, RB, F, AG, AR, U, AU, MO, SE, GD,
YB, VX, SM, LI, RE, N, W, OS, HO, PI,
EDO, PG4, OGA, SO4, HEZ, FEO, CL, DMS,
ACT, MPD, GOL, NH2, CUA, SIW, PGW, IOD,
BR, 3NI, ZRW, 78M, UNX, MES, CCN, PO4

(which have up to 3 letters) from the PDB corresponding to such non-biological molecules based on the BioLip database [141] and our own manual inspection as shown in Table 2.1. For example, GOL is the three letter code for glycerol, a common solvent.

We remove all non-biological from our training set as they generally represent non-specific binding partners and/or molecules held in place by a crystal lattice rather than by protein interactions. We note that the list in Table 2.1 may not be exhaustive, but it filters out the most common examples we observed in the PDB.

Contacting Chains A protein chain is considered "in contact" with the query ligand if at least 5 Cα atoms of that protein chain are within 30Å of any atom in the query ligand or if any atom in the protein chain is within 5Å of any atom in the query ligand. This definition of contacting protein chain allows for a broad biological context in which to predict the query ligand of interest.

A non-polymer chain is considered "in contact" with the query ligand if at least one atom of that chain is within 5Å of the query ligand, or if all atoms are within 30Å of the query ligand. The model thus learns to predict not only the relative positions of a ligand and its protein binding partners, but also associated cofactors in the same or nearby binding pockets.

Covalent Modification Dataset We separate query ligands that are covalently bonded to a protein into a distinct and separate dataset. For each such query ligand, we filtered out

Table 2.2: Selected Metals

LA, NI, 3CO, K, CR, ZN, CD, PD, TB, YT3, OS, EU, NA, RB, W, YB, HO3, CE, MN, TL, LI, MN3, AU3, AU, EU3, AL, 3NI, FE2, PT, FE, CA, AG, CU1, LU, HG, CO, SR, MG, PB, CS, GA, BA, SM, SB, CU, MO, CU2

those that involved a covalent bond between an oxygen atom on the ligand and an oxygen atom on the protein, as well as protein-ligand fluorine-flourine bonds, covalent bonds to hydrogen atoms, and all cases where the length of the protein-ligand covalent bond was less than 1Å. We find that such covalent protein-ligand bonds are usually between a protein and a non-biological small molecule, as described in subsection 2.10.1.

Metal Ion Dataset We also separate query atoms that represent metal ions into a distinct dataset. In order to avoid expanding the vocabulary set of our model too greatly and potentially introducing non-specific metal binding sites into the dataset, we only train on the metals whose 3-letter PDB codes are listed in Table 2.2.

Small Molecule Datasets Every remaining query ligand can be grouped into one of the following groups: multi-residue, multi-protein assembly, and single-protein assembly. We describe the three sub-datasets here:

1. **Protein/Small Molecule Complex**: Dataset containing all single residue small molecules that only interact with a single other protein chain. However, the query ligand in each item of this dataset lies on a single residue in the PDB entry.

2. **Protein/Multi-Residue Ligand Complex**: Dataset containing all "multi-residue" ligands. Such ligands exist as multiple residues (or a single residue with multiple bioassembly transforms) in their respective cif file, and usually represent sugar chains or small peptides.

3. **Protein/Small Molecule Assembly**: Dataset containing non-protein biomolecular

Table 2.3: Number of Protein Sequence Clusters In Each PDB Training Dataset

Dataset	**Sequence Clusters**	**Examples**
Protein Monomer	21,648	301,934
AF2 Distillation Set	1,036,080	3,605,951
Protein Heteromer	13,755	183,821
Protein Nucleic Acid	1,235	17,240
RNA	1,449	6,522
Protein Small Molecule	5,662	121,800
Protein Metal Complex	5,324	112,456
Protein Multi-Residue Ligand	613	4,775
Protein Small Molecule Assembly	2,564	43,838
Covalent Modification	1,099	12,689

context (small molecules, covalent modifications, multi-residue ligands, metal ions) and > 1 protein chain. This dataset is distinct from 1. because we generate paired MSAs when there are two or more distinct (heteromeric) protein chains in a complex.

Dataset Clustering We cluster each entry in each dataset by their *primary protein partner*. For protein-only datasets, this is either the protein monomer, or in the case of hetero-oligomers, we arbitrarily select the first chain that appears in the PDB entry as the primary chain. For nucleic acid and small molecule datasets, we designate as the primary protein partner the protein chain with the greatest number of atoms within 5Å of the query nucleic acid/ligand. All items in each database are then clustered using MMSeqs2 [116] using the default hyper-parameters and subsequently used for dataset sampling at training time.

Final Statistics After all the filters were applied, we kept chains that have resolution < 4.5Å and that were deposited in the PDB before April, 30th, 2020 for the purposes of held-out evaluation on PDB entries from 2021 onward. 10% of protein sequence clusters were held out of training for validation during training. The final number of items and clusters in each dataset is shown in Table 2.3. The number of validation items and clusters in each dataset is shown in Table 2.4.

Table 2.4: Number of Protein Sequence Clusters In Each PDB Validation Dataset

Dataset	**Sequence Clusters**	**Examples**
Protein Monomer	1,206	47,094
Protein Homomer	682	25,284
Protein Heteromer	1792	24,454
Protein Nucleic Acid	79	1,272
RNA	153	720
Protein Small Molecule	354	18,110
Protein Multi-Residue Ligand	51	302
Protein Small Molecule Assembly	163	6,165
Covalent Modification	80	1,834

CSD Dataset

In addition to molecules in the PDB, we augment our training dataset with small molecule crystal structures from the Cambridge Structure Database (CSD v5.43; November 2021) [54]. We filter structures based on the following metrics: 1) resolution < 5Å, 2) not polymeric, 3) greater than 5 atoms resolved, 4) less than 100 atoms resolved, and 5) ability to be parsed by OpenBabel [93]. We sample molecules with equal probability and separate a validation set that does not have Tanimoto score >0.75 to any molecule in training.

Negative Datasets

Following RF2 and RF2NA, we use a set of "negative" interactions to help the network focus on relevant features that constitute binding. Briefly, for proteins this means showing examples where we randomly pair chains and only assess a loss on each individual chain and for nucleic acids it involves mutating bases that make essential contacts for the formation of the complex. Negative examples were not shown for any small molecule dataset.

Table 2.5: Element Tokens in RFAA

Al, As, Au, B, Be, Br, C, Ca, Cl,
Co, Cr, Cu, F, Fe, Hg, I, Ir, K, Li,
Mg, Mn, Mo, N, Ni, O, Os, P, Pb, Pd,
Pr, Pt, Re, Rh, Ru, S, Sb, Se, Si,
Sn, Tb, Te, U, W, V, Y, Zn

2.10.2 Modeling Arbitrary Biological Inputs

Architectures for modeling both protein structures and nucleic acid structures have been previously described. In this subsection, we describe the necessary changes to such architectures to model small molecules, covalent modifications to protein structures and arbitrary non-canonical amino acids.

Expanded Input Sets

The most significant architectural change from existing protein structural networks is the expanded input features that RFAA takes in, which we describe in the following subsections.

New Tokens The original RoseTTAFold architecture had 22 tokens: 20 amino acids, 1 unknown and 1 gap token. The RF-Nucleic Acid expanded this token set by 10, adding 8 distinct tokens for the 4 DNA and 4 RNA bases, and 2 tokens for unknown DNA base and unknown RNA base, respectively. The RFAA architecture includes 46 additional tokens representing individual atoms with the element types shown in Table 2.5, a token for deprotonated histidine (unused in practice, left in for legacy reasons) and an unknown atom token for a total token count of 80.

Bond Connectivity When predicting structures of arbitrary molecules, it is important for the network to know the bond connectivity of those molecules. To provide this information to the network, we pass in the bond connectivity of input molecules as a 2D bond adjacency matrix as an input. We designate 7 bond types representing single bond, double bond, triple bond, aromatic bond, residue-residue (or base-base), residue-ligand atom bond and other

Table 2.6: Bond Types in RFAA

Bond type	**Encoding**
No bond	0
Single	1
Double	2
Triple	3
Aromatic	4
Protein residue-residue or nucleic acid base-base	5
Residue-atom	6
Other	7

bond type. In practice, the "other" bond type is not used but exists for historical reasons. The residue-residue "bond" type exists to be able to provide bond features for protein and nucleic acid inputs, so that the input bond matrix is always of the same dimension as the input. Residue-atom bonds exist in order to do a process we call residue *atomization*, which is used to model ligands that are covalently bonded to a residue and arbitrary non-canonical amino acids. This process is described further in subsection 2.10.3.

Chiral Features Another key bias in more generalized biomolecular modeling is chirality. Aforementioned features like atom types and bond connectivities are not sufficient to specify the chirality of input molecules, so we provide chiral features explicitly to the network at each chiral center. In this work, we only deal with tetrahedral chirality and leave more complicated forms of stereochemistry to future work.

For each tetrahedral chiral center, we enumerate all sets of three heavy atom neighbors in all orders. For each ordering, we compute the pseudo-dihedral angle between those four points (center and 3 heavy atoms) and note whether that angle is positive or negative, which determines the chirality of the center uniquely. For ideal tetrahedral geometry, the magnitude of the dihedral angle is $\arcsin(1/\sqrt{3})$. See subsection 2.10.4 for further details.

We compute the difference between the predicted dihedral angle and the ideal dihedral angle of the chiral center, and pass the coordinate-wise gradients of the difference as an input to

the 3D track of the network. This representation of chirality provides direct signal to the network on how to update atomic coordinate positions in order to obey the ideal tretrahedral geometry. This process is described further in subsection 2.10.4.

Atom Frames Key to the success of AF2 and RF2 is the frame aligned point error loss [72]. This involves aligning N-Cα-C backbone frames of predicted structures to true structures and then measuring the error of all the other predicted atoms with respect to the true structure in that alignment. This loss has attractive properties for biomolecules such as not being invariant to reflections which allows the network to predict correct chirality. We construct *canonical* frames for each atom in small molecules comprising of atoms and their bonded neighbors. We achieve this by iterating through all bonded triplets of atoms and assigning each triplet a priority based on the bonded atoms, depicted in Table 2.7. The process for constructing canonical frames from a ligand is outlined below:

1. Construct a graph where each node is an atom and each edge is a bond.

2. For each atom in the graph:

   (a) Enumerate through all paths of length three containing that atom. If there exist paths such that the given atom is in the center, exclude all other paths.

   (b) Compute frame priorities for each atom in each path and make a list of frame priorities in increasing order.

   (c) For each such path of length three, sort them by lexicographic atom frame priority, e.g. for two paths A and B, path A will appear before path B if and only if either the lowest frame priority in path A is less than the lowest frame priority in path B, or they are equal and the second lowest frame priority in path A is less than the second lowest in path B, and so on.

   (d) The first path in the lexicographic order is chosen as the frame for this atom, and the order of the frame is determined by frame priority in increasing order.

This process deterministically computes a local coordinate frame for each atom in arbitrary molecules (with at least 3 atoms). If a frame has an unresolved atom, it still is assigned as the canonical frame but is not used in loss calculation. The usage of the atom frames is further discussed in subsections 2.10.4, 2.10.5 and 2.10.5. Importantly, these features are constructed from the bond graph so they maintain the permutation invariance of the inputs to the network.

Element Type	**Priority**
K	0
Li	1
Ca	2
Mg	3
Be	4
Y	5
Tb	6
U	7
V	8
W	9
Mo	10
Cr	11
Re	12
Mn	13
Os	14
Ru	15
Fe	16
Pr	17
Ir	18
Rh	19
Co	20

Pt	21
Pd	22
Ni	23
Au	24
Cu	25
Hg	26
Zn	27
Al	28
B	29
Pb	30
Sn	31
Si	32
C	33
Sb	34
As	35
P	36
N	37
Te	38
Se	39
S	40
O	41
I	42
Br	43
Cl	44
F	45

Table 2.7: Frame Priorities of Atoms in RFAA

Positional Encodings To break permutation symmetry for sequences, we use a signed relative positional encoding for protein sequences and nucleic acid bases [72]. Atomic graphs require permutation symmetry so we do not provide a relative positional encoding. For atomic inputs, we provide a separate embedding that measures the shortest distance between any pair of atoms in the bond graph described in subsection 2.10.2. We develop a generalization of these two embeddings for cases where atom nodes are bonded to residues to encode the distance between an atom and its closest bonded residue. Further details are provided in subsection 2.10.4.

2.10.3 Data Pipeline

Our data pipeline involves taking raw data from cif files and formatting the data into input tensors for the network. We first will describe the inputs to the network and then go through details of how each dataset is preprocessed. We use OpenBabel to parse the ideal sdf files for each PDB but do not use any chemical quantities computed by it (just element types, bond types and whether an atom is chiral center). We find this to be preferable because OpenBabel can process all ideal sdf files provided by the PDB so we do not exclude examples because of parsing errors. We compute the direction of the chiral center as discussed in subsection 2.10.4.

Inputs for RFAA

Remaining features such as MSAs and templates are handled identically for proteins to RF2. The coordinate dimension, 36, reflects the maximum amount of heavy atoms and hydrogens possible in a residue or base. The small molecule tokens are appended to the first sequence in all the MSA features and the remaining MSA sequences are initialized with gap tokens. Small molecules receive empty template features which are concatenated block diagonally to the protein features. A detailed description of the inputs are shown in Table 2.8.

Input (dimension)	**Description**

msa_masked ($N_{num\_clusters}$, L, 164)	Clustered MSA with some portions of the sequences masked. For atom nodes, the first sequence has its respective atom tokens and then remaining sequences are filled with gap tokens. (80 raw msa, 80 cluster statistics, 2 insertions/deletions, 2 Nterm/Cterm)
msa_full ($N_{num\_sequences}$, L, 80)	Full MSA clipped at 1024 sequences.
seq (L, 80)	First row of the MSA. In this case, the protein sequence and any atom tokens, including mask tokens.
idx (L)	Residue index of each residue in the input. This input must be provided for atom nodes but has no semantic meaning (it is unused by the network).
bond_feats (L, L, 7)	Pairwise bond adjacency matrix. Pairs of residues are either single, double, triple, aromatic, residue-residue, residue-atom or other.
dist_matrix (L, L)	Minimum amount of bonds to traverse between two nodes. This is 0 between all protein nodes.
chirals ($L_{num\_chiral\_centers}$, 5)	All orderings of 4 atoms around a chiral center (first four dimensions) and the ideal pseudo-dihedral angle formed by that ordering of atoms (fifth dimension).

atom_frames (L_{num_atoms}, 3, 2)	Indices that form frames for each atom node in the input. The second dimension represents that there are three atoms in each frame. The third dimension represents an offset in the node dimension because atom frames go across nodes and the absolute index in the atom dimension.
t1d ($N_{num\_templates}$, L, 80)	1D template feature. First, 79 represent the "sequence" (residue/atom types) of the templated structure. Last dimension represents residue wise template confidence.
t2d ($N_{num\_templates}$, L, L, 64)	2D template information which gives the binned distances and angles between frames (N-Cα-C for proteins, designated atom frame for atoms)
alpha_t ($N_{num\_templates}$, L, 30)	Sidechain torsion angles from templates (10 angles x sin, cos and whether the angle exists in the structure for each residue)
msa_prev ($N_{num\_clusters}$, L, C_m)	Recycled MSA features. C_m=256 (number of 1D channels)
pair_prev (L, L, C_p)	Recycled pair features. C_p=192 (number of 2D channels)
state_prev (L, C_s)	Recycled state features. C_s=32 (number of 3D $\ell 0$ channels)
xyz_prev (L, 36, 3)	Recycled XYZ coordinates. On first iteration, this is set to the coordinates from the first template. If no templates, coordinates are initialized at the origin with random noise (between -2.5 and 2.5Å) applied.

sc_torsions_prev (L, 30)	Recycled predicted sidechain torsion angles.

Table 2.8: Inputs to RFAA

Featurization of Symmetric Permutations

When featurizing multimers (empirically in the homomer and small molecule assembly datasets), there are often identical chains that, if swapped, result in the identical complex. We want the gradient of the loss to push the network towards the relabeled complex that is closest to the prediction so during preprocessing we track which chains can be swapped so that we can deconvolute which permutation to apply the loss on during training. We use the same scheme to account for permutation swaps of atoms in small molecule structures.

Featurization of Protein Small Molecule Complexes

A similar preprocessing procedure was followed for the small molecule protein complex dataset, the multichain residue ligand dataset, the covalent modification dataset and the protein-small molecule assembly dataset (multiple contacting protein chains). Each training example centers around a single query molecule in a specific bioassembly. Based on the details of the bioassembly features for the nearest chains (both protein and other small molecules) are constructed. There are two types of biomolecular contexts that are sampled stochastically. First, metal ions in the presence of other small molecules are sampled stochastically because often is is not a priori known when solving a structure whether there will be a metal crystallized in the pocket. Second, if there modified residues present in the cif file, they are featurized as an atomized residues rather than their canonicalized version.

Due to memory restrictions, we then perform a cropping procedure to select a subset of nodes to represent a training example. The cropping procedure samples a random atom on the query molecule and computes the distance to all other atoms or Cα atoms in proteins. It then selects the top n_crop nodes to include in the crop. In our early experiments we

found that sometimes, there were protein chains that were either too short or far away from the ligand in euclidean space that were included in the crop with insufficient context to be predicted accurately. In these cases, the gradient was dominated by the incorrect prediction of those protein chains and not on the correct docking of the small molecule. To remedy this, after finding the top n_crop nodes, we iterate through all the protein chains that were in the crop and remove any chains with <10 contacts to other nodes in the crop or with less than 8 residues. After these chains are removed, we noticed that certain molecules in the crop also did not have sufficient contacts to be docked so we iterate through all the molecules in the crop and remove any molecules that have <4 contacts to a protein chain. The exact logic is shown in 1.

It is evident from this cropping process that full subunits in large symmetric assemblies could be cropped out. Since we compute potential symmetric relabeling of chains before cropping (see subsection 2.10.3), certain chains that were computed as potential symmetric relabelings are no longer valid (specifically because the small molecule context should drive the network to predict a specific interface when a symmetric oligomer could have multiple distinct protein-protein interfaces). After cropping, we reiterate through the precomputed symmetric permutations and remove those that are no longer possible given the chains that were removed during cropping.

Featurization of Atomized Protein Examples

Training examples for *atomized* proteins first are featurized identically to protein monomer examples (except the stochastic homomer featurization is turned off, see subsection 2.10.6). After cropping, a number of residues is sampled from Uniform(3,5) and that number of (fully resolved) contiguous residues is chosen for atomization. If there are not enough valid residues in the crop to *atomize*, we treat the example as a monomer example. We then take that selection of residues, featurize them using all the small molecule features (atom tokens, bond features, chirality inputs). We also provide bond tokens to indicate bonds between the first N token in the *atomized* region to the previous residue and the last C token to the following residue. Finally, the MSA and template information for these residues is removed from the

Algorithm 1 Cropping for SM Complex Datasets

function CROP_SM_COMPL($xyz, query\_mol, n_{crop}$)
  $atom_{query} \leftarrow Uniform(atoms_{query\_mol})$
  $d = ||xyz_{atom_{query}} - xyz_j||$ ▷ distances between query atom and all Cαs
  keep $\leftarrow n\_lowest\_values(n = n_{crop}, \mathbf{D})$
  for chain=$0 \ldots N_{sm\_chains}$ **do**
    if any(keep *in* chain) **then**
      keep += all(atoms *in* chain) ▷ do not crop ligands
    end if
  end for
  keep_atoms $\leftarrow$ atoms *in* keep
  for chain=$0 \ldots N_{protein\_chains}$ **do**
    keep_chain $\leftarrow$ keep *in* chain
    $d_{ca} = ||xyz_{keep\_chain} - xyz_{keep\_atoms}||$
    if Count(keep_chain) < 8 OR Count($d_{ca} < 4$) < 10 **then**
      keep -= keep_chain ▷ remove chains with few contacts to small molecule
    end if
  end for
  keep_residues $\leftarrow$ residues *in* keep
  for chain=$0 \ldots N_{sm\_chains}$ **do**
    keep_atom_chain $\leftarrow$ keep *in* chain
    $d_{ar} = ||xyz_{keep\_atom\_chain} - xyz_{keep\_residues}||$
    if Count(d_{ar} <4) < 4 **then**
      keep -= keep_atom_chain ▷ remove small molecules with no contacts to proteins
    end if
  end for
  return keep
end function

input features so the network must learn how to generate their structures and poses from the atomic information. Practically, we precompute the atoms, bonds and chiralities of each atom in each residue and convert the features as shown in Algorithm 2. Symmetric swaps of sidechain atoms are accounted for in the same manner as subsection 2.10.3.

Algorithm 2 *Atomization* of Protein Residues

```
function ATOMIZE_PROTEIN(residue_indices, seq, *protein_features)
    atoms ← CONCAT(atoms_in_residues(seq[residue_indices]))
    bonds ← BLOCK_DIAGONAL(bonds_in_residues(seq[residue_indices]))
    bonds += atomized_peptide_bonds          ▷ add peptide bonds between adjacent atomized residues
    chirals ← chirals_per_residue(seq[residue_indices])
    frames = get_atom_frames(atoms, bonds)
    DELETE *protein_features[residue_indices]
end function
```

Featurization of Covalently Bound Ligands

Covalently bound ligands are preprocessed very similarly to small molecule complexes. The covalent modifications are modelled just as other ligands would be. The residue that has the covalent bond to the ligand is atomized using the same method as subsection 2.10.3. The atom in the ligand and the atom in the *atomized* residue is provided in the bond features (eg. single bond between atom i from modification and atom j in atomized residue). All other featurization (protein MSA, templates etc) remains the same as other datasets.

Featurization of Metal Ions

Metal Ions are provided to the network as a single atom ligand. The only difference is that since metal ions only have a single atom, they do not have their own canonical frame. In these cases, the network does not receive a frame input and there is no loss calculated with respect to the frame of the ion (there are still gradients from the error of the placement of the ion with respect to the other frames in the structure).

Featurization of CSD Small Molecule Crystals

Asymmetric units of crystal lattices from the CSD are featurized identically to small molecules that are bound to proteins. The network is then tasked with predicting the atomic coordinates of the molecule. Molecules with less than 5 atoms, greater than 100 atoms, polymers or resolution >5Å are discarded from the dataset. Remaining molecules that could be parsed by OpenBabel were used to train the network.

2.10.4 Algorithm Details

The architecture of RFAA is similar to RF2. The network accepts 1D, 2D and 3D inputs and treats structure prediction as a graph inference problem where the objective is to find edges in the graph where edges are distances in euclidean space. The outputs of the model are predicted coordinates, confidence measures and the latent embeddings from the three tracks. There are 3 main stages of the network, the embedding stage, the simulator stage and the refinement layers. Similar to AF2 and RF2, the network employs recycling where latent features (shown in Algorithm 3) from previous forward passes to the network are provided to future iterations. Here, we will mainly cover the places where our implementation diverges from RF2. Importantly, the RFAA architecture has three main features that are passed throughout the network, *msa*, *pair* and *state*. *msa* are features from the 1D track, *pair* are features from the 2D track and *state* are node-wise features from the 3D track. The outputs of the network include *xyz* and *sc_ torsions* which are the predicted backbone coordinates and predicted side chain torsion angles respectively (which can be used to construct the full atom coordinates). For clarity, intermediate outputs of the network are in italics and preprocessed features are in plaintext.

Similar to RF2, the first four blocks process the *msa_ full* features. In these layers, the

Algorithm 3 Recycling Iterations

msa, pair, state = **0**, **0**, **0**
xyz, sc_ torsions = Uniform(-2.5,2.5), **0** ▷ or template structure if available
for i=0...N_{cycle} **do**
    msa, pair, state, xyz, sc_ torsions = RFAA(*input_ features, *msa, pair, state, xyz, sc_ torsions*)
end for

network uses global column attention over the full MSA instead of column attention and biased row attention used in the 36 main blocks. This portion of the network is meant to extract extra information for cases with deep MSAs. Similar to all blocks in RFAA, there are predicted structures generated for each block and the features from those bias the future interactions of the network. The implementation of these are unchanged from RF2 (except the additional atomic context is also processed), the intuition being that these layers are

mainly for MSA processing that is not applicable to atomic graphs.

The next 32 layers are the main blocks of the network. They process the *msa_masked* features along with the biomolecular context. The main block logic is shown in Algorithm 5. Each block in RFAA has five steps, first the computation of the structure bias from the

Algorithm 4 RFAA Forward Pass

function RFAA(msa_masked, msa_full, seq, idx, bond_feats, dist_matrix, chirals, atom_frames, t1d, t2d, alpha_t, msa_prev, pair_prev, state_prev, xyz_prev, sc_torsions_prev, N_{full_blocks}=4, N_{main_blocks}=32, N_{ref_blocks}=4)
    msa_{full} = MSA_Full_Embed(msa_full)
    $msa_{cluster}, pair, state$ = MSA_Cluster_Embed(msa_masked, seq, idx, bond_feats, dist_matrix)
    $pair$ += Bond_Embed(bond_feats)
    $msa_{cluster0}, pair, state$ += Recycle_Embed(msa_prev, pair_prev, state_prev, xyz_prev, sc_torsions) ▷ Only update first row of MSA
    $pair, state$ += Template_Embed(t1d, t2d, alpha_t, $pair$, $state$)
    for i=0...N_{full_blocks} **do**
        Stopgrad(xyz)
        $msa_{full}, pair, state, xyz, sc\_torsions$ = Full_Block(msa_full, $pair$, $state$, xyz)
    end for
    for i=0...N_{main_blocks} **do**
        Stopgrad(xyz)
        chiral_grads = calc_chiral_grads(xyz, chirals)
        $msa_{cluster}, pair, state, xyz, sc\_torsions$ = Main_Block($msa_{cluster}, pair, state, xyz$, idx, bond_feats, dist_matrix, chiral_grads, atom_frames)
    end for
    for i=0...N_{ref_blocks} **do**
        Stopgrad(xyz)
        chiral_grads = calc_chiral_grads(xyz, chirals)
        clash_l1_grads, clash_l0_grads = calc_clash_grads(xyz, $sc\_torsions$)
        $state, xyz, sc\_torsions$ = Ref_Block($msa_{cluster}, pair, state, xyz$, idx, bond_feats, dist_matrix, chiral_grads, clash_l1_grads, atom_frames, clash_l0_grads)
    end for
    return $msa_{cluster}, pair, state, xyz, sc\_torsions$
end function

previous block, second the *msa* feature update, third the update of the *pair* features based on the *msa* features, fourth the update of the *pair* features and finally the update of the 3D features. The full algorithm is shown in Algorithm 4, which has 83M parameters that are

optimized through training.

Algorithm 5 RFAA Main Block

function MAIN_BLOCK(*msa_cluster*, *pair*, *state*, *xyz*, idx, bond_feats, dist_matrix)
    str_bias = RBF($||xyz_{C\alpha} - xyz_{C\alpha}||$)
    rel_pos, bond_dist = Positional_Encoding(idx, bond_feats, dist_matrix)
    str_bias += Linear(rel_pos) + Linear(bond_dist)
    $msa_{cluster}$ = 1D_Update($msa_{cluster}$, *pair*, *state*, *str_bias*)
    pair = Aggregation(*msa_cluster*, *pair*)
    pair = 2D_Update(*pair*, *str_bias*, *state*)
    $xyz, state, sc\_torsions$ = 3D_Update($msa_{cluster}$, *pair*, *state*, *xyz*, idx, bond_feats, dist_matrix, atom_frames)
    return $msa_{cluster}, pair, state, xyz, sc\_torsions$
end function

1D Embeddings

The 1D embeddings are very similar to those in RF2. For the *msa_full* blocks, they are identical. The msa_full and seq input features are embedded and the embedding of the *seq* features are added to every row of the *msa* features. The *msa_masked* embedding slightly diverges from RF2 (6). The *msa_masked* embedding begins with the exact same operations as the *msa_full* embedding that construct the initial *msa* features. Then, the initial *pair* and *state* features are constructed. The *pair* features are constructed by embedding the *seq* input with two sets of weights and then computing the outer product of the two embeddings.

At this point the relative positional encoding is added (in the *msa_full* blocks the attentions are column-wise so breaking row-wise symmetry is less important). Since atoms nodes in small molecules are permutation invariant, we remove the positional encoding for these nodes. When considering *atomized* residues which are in the polypeptide chain, we must break permutation symmetry for these atoms. We intended to provide the network both information about where the atoms are in the polypeptide chain and the relative distances within the bond graph between atoms (and between atoms and residues). We devised two encodings, the first is the signed relative position encoding (capped between -32 and 32) similar to RF2 and the second is a relative bond graph distance (capped at 8). For each

atom in the bond graph we compute the shortest path through the bond graph to all other atoms. For cases where atoms are bonded to residues, we count the atom-residue bond as an additional bond in the shortest distance feature and then provide the closest residue's relative positional offset. Both of these features are linearly embedded and summed (Algorithm 7). Finally, the *state* features are initialized by embedding the *seq* inputs.

Algorithm 6 Clustered MSA Features Embedding

function MSA_Cluster_Embed(msa_masked, seq, idx, bond_feats, dist_matrix)
    $msa_{cluster}$ = Linear(msa_masked)
    seq = Linear(seq)
    $msa_{cluster0}$ += seq
    seq_i, seq_j = Linear(seq) ▷ Different weights
    $pair$ = Outer_Product(seq_i, seq_j)
    rel_pos, bond_dist = Positional_Encoding(seq, idx, bond_feats, dist_matrix)
    $pair$ += Linear(one_hot_encode(rel_pos))
    $pair$ += Linear(one_hot_encode(bond_dist))
    $state$ = Linear(seq)
    return $msa_{cluster}, pair, state$
end function

Algorithm 7 Generalized Positional Encoding

function Positional_encoding(idx, bond_feats, dist_matrix)
    rel_pos = clip($idx_i - idx_j$, -32, 32)
    bond_dist = clip(dist_matrix, 8)
    if atom_residue_bond **then** ▷ Atomized protein or covalent modification
        closest_termini = argmin(dist_matrix[:, terminal_N, terminal_C])
        closest_residues = bonded_residue(bond_feats, closest_termini)
        rel_pos[atomized_nodes] ← rel_pos[closest_residues] ▷ add relative position of closest residue for each atomized nodes
    end if
    return rel_pos, bond_dist
end function

Bond Embedding

We reasoned that bonds provide very straightforward constraints on pairwise distances between atoms. We linearly embed the one-hot encoded bond graph and sum it with the *pair* features to allow the network to predict accurate bond lengths and angles.

Algorithm 8 Bond Embedding

```
function BOND_EMB(bond_feats)
    bond_emb = Linear(one_hot_encode(bond_feats))
    return bond_emb
end function
```

1D Track Update

The 1D track update is identical to the 1D track update in RF2. The *pair* features are passed through a LayerNorm and then an embedding of the radial basis function (64 bins, 0.5 Å increments, standard deviation=0.5) of the distances from the predicted coordinates is added to *pair* features. A LayerNorm is applied to the *state* features and then it is embedded. The first row in the *msa* embedding (representing the query sequence embedding) is updated with the embedded state features. The *msa* embedding is then updated with row attention with bias from the *pair* features and then column-wise attention is computed to update the *msa* features [11, 72, 103].

2D Track Update

The 2D track update is identical to the update in RF2. The *pair* features are updated using the TriangleMultiplicationOutgoing and TriangleMultiplicationIncoming updates with 0.25 probability of dropout for each. We then apply structure biased axial attention as in RF2. To form the structure bias, the distances between Cα (or P for nucleic acids and atom for atom nodes) coordinates predicted at the previous block are binned with gaussian blur using a radial basis function (64 bins, 0.5 Å increments, standard deviation=0.5). This binned distribution is embedded to match the number of channels in the pair dimension. To gate which distances should bias the pair features, the *state* features are embedded with two separate sets of weights and the outer product of the embeddings is computed. The outer product goes through another linear embedding, followed by a *sigmoid* activation function. This final value is used as a gate and multiplied by the binned distance distribution which represents the coordinate bias that will be applied. Row and column attention are computed for the pair feature and the coordinate bias is added to the attention weights. Finally, a

FeedForward layer updates the final pair representation from the block.

We did not change the functional form of the 2D update from RF2 but we expected that the bottleneck to learning would be the 2D update which has to learn many new types of interactions about arbitrary biomolecules. We expected that a more rich pair representation and more attention heads would give the network enough capacity to learn the new interactions that were shown in the training set following intuition from [72] and [20]. We increased the number of pair channels to 192 (from 128 in RF2) and the number of attention heads per update to 6 (from 4 in RF2).

3D Track Update

The 3D track update is where our implementation diverges the most from RF2. The goal of the 3D track update is to use the features from the 1D and 2D tracks and predict the coordinates of the complex. Similar to RF2, we sought to use the SE(3)-Transformer architecture which guarantees equivariance over the group of rotations and translations by projecting features into a basis formed by the spherical harmonics [50, 120]. The SE(3) Transformer is a graph neural network which takes in node features for every node and edge features for each connected pair of nodes and aggregates features across them. This architecture uses tensor products to mix $\ell 0$ (scalar features), $\ell 1$ features (vector features) and higher order ℓ features and can predict arbitrary ℓ features for each node. In this subsection, we will describe how we formulated $\ell 0$ and $\ell 1$ features for each node that is modelled and edge features between nodes. The full algorithm is shown in Algorithm 9.

Constructing $\ell 0$ features The first row of the *msa* features (the features corresponding to the query sequence) and the *state* features are passed through a LayerNorm. These values are concatenated and then processed by a shallow network consisting of a Linear embedding, a FeedForward Layer and a LayerNorm. These features are all scalar values and represent the $\ell 0$ node features.

Providing the Network Chirality Inputs To provide the network chirality features, we aimed to construct a geometric representation of chirality that does not involve learning

Algorithm 9 RFAA Structure Update

function 3D_UPDATE($msa_{cluster}, pair, state, xyz, idx, bond\_feats, dist\_matrix, chiral\_grads, atom\_frames$)
    seq = LayerNorm($msa_{cluster0}$) ▷ Query sequence latent embedding
    $pair, state$ = LayerNorm($pair$), LayerNorm($state$)
    $node\_feats$ = Linear(CONCAT([$seq, state$]))
    $node\_feats$ += FeedForward($node\_feats$)
    $node\_feats$ = LayerNorm($node\_feats$)
    rel_pos, bond_dist = Positional_Encoding(idx, bond_feats, dist_matrix)
    rbf_feat = rbf($||xyz_{C\alpha} - xyz_{C\alpha}||$) ▷ Or P or atom coordinate
    $edge\_feats$ = Linear(CONCAT([$pair$, rbf_feat, rel_pos, bond_dist]))
    $edge\_feats$ += FeedForward($edge\_feats$)
    $edge\_feats$ = LayerNorm($edge\_feats$)
    $frame_{ij}, frame_{kj} = frame_i - frame_j, frame_k - frame_j$ ▷ Construct frames from N-Cα-C, O-P-O or atom_frames
    l1_feats = CONCAT([$frame_{ij}, frame_{kj}, chiral\_grads$]
    geometric_feats = $xyz_i - xyz_j$
    $state, coord\_update$ = SE3_Transformer(node_feats, edge_feats, l1_feats, geometric_feats)
    $quaternion\_values, T = coord\_updates$
    R = construct_rotation_matrix_from_quaternion($quaternion\_values$) ▷ Set R = Identity matrix for "atom" nodes
    $xyz = R \circ xyz + T$
    $sc\_torsions$ = SC_pred($msa_{cluster0}, state$)
    return state, xyz, sc_torsions
end function

human made concepts such as (r) and (s) (as the network would need to learn how to implicitly order substituents and determine whether they were presented in the clockwise or counterclockwise order). We observed that set of atoms in chiral centers form planes and the angle between the frames are different depending on the stereochemical identity of that center. We then construct an ideal tetrahedron on the unit sphere with centroid at the origin (where the chiral center would be), which composes of four points:

$$v_1 = (\sqrt{\frac{8}{9}}, 0, -\frac{1}{3})$$
$$v_2 = (-\sqrt{\frac{2}{9}}, \sqrt{\frac{2}{3}}, -\frac{1}{3})$$
$$v_3 = (-\sqrt{\frac{2}{9}}, -\sqrt{\frac{2}{3}}, -\frac{1}{3})$$
$$v_4 = (0, 0, 1)$$

We can then compute the dihedral angle between the planes formed by (v_1, v_2, v_3) and (v_2, v_3, v_4):

$$\begin{aligned}
&\text{Dihedral}(a, b, c, d) = \text{atan2}(\\
&\qquad [c-b] \cdot (([b-a] \times [c-b]) \times ([c-b] \times [d-c])), \\
&\qquad \|c-b\|([b-a] \times [c-b]) \cdot ([c-b] \times [d-c])) \\
&\text{Dihedral}(v_1, v_2, v_3, v_4) = 1.23 \,\text{radians}
\end{aligned}$$

In the example above, a switch in stereochemistry would result in the following coordinates(v_3 comes out of the plane and v_4 goes into the plane):

$$v_1 = (\sqrt{\frac{8}{9}}, 0, -\frac{1}{3})$$
$$v_2 = (-\sqrt{\frac{2}{9}}, \sqrt{\frac{2}{3}}, -\frac{1}{3})$$
$$v_3 = (0, 0, 1)$$
$$v_4 = (-\sqrt{\frac{2}{9}}, -\sqrt{\frac{2}{3}}, -\frac{1}{3})$$
$$\text{Dihedral}(v_1, v_2, v_3, v_4) = -1.23 \text{ radians}$$

The positions of these four atoms are sufficient to determine chirality. The dihedral angle between planes (v_1, v_2, v_3) and (v_2, v_3, v_4) will be positive in the first case and negative in the second. In practice in RFAA, we do not always have access to all four substituents of a chiral center (one of them could be a hydrogen which is not explicitly modelled). Despite this we do have sufficient information to determine the chirality of a given system given three points since we know the chiral center has coordinates: $o = (0, 0, 0)$. We can then construct planes consisting of (o, v_1, v_2) and (v_1, v_2, v_3), and compute the dihedral angle between them which will be either $\arcsin \frac{1}{\sqrt{3}}$ or $-\arcsin \frac{1}{\sqrt{3}}$ (0.6155 radians or -0.6155 radians).

$$o = (0, 0, 0)$$
$$v_1 = (\sqrt{\frac{8}{9}}, 0, -\frac{1}{3})$$
$$v_2 = (-\sqrt{\frac{2}{9}}, \sqrt{\frac{2}{3}}, -\frac{1}{3})$$
$$v_3 = (0, 0, 1)$$
$$\text{Dihedral}(o, v_1, v_2, v_3) = 0.6155 \text{ radians}$$

To show that this angle has enough information to convey information about chirality, we invert the chirality of the system and compute the angle:

$$o = (0, 0, 0)$$
$$v_1 = (\sqrt{\frac{8}{9}}, 0, -\frac{1}{3})$$
$$v_2 = (-\sqrt{\frac{2}{9}}, \sqrt{\frac{2}{3}}, -\frac{1}{3})$$
$$v_3 = (-\sqrt{\frac{2}{9}}, -\sqrt{\frac{2}{3}}, -\frac{1}{3})$$
$$\text{Dihedral}(o, v_1, v_2, v_3) = -0.6155 \text{ radians}$$

There are two pieces of information necessary to compute these angles: the two planes to consider (an ordering of four atoms) and the resultant ideal angle. It is not clear how to embed this information into a neural network such as RoseTTAFold (how do we embed the order of the indices used to compute the angles?) so we decided to take inspiration from physical modeling where gradients of energy functions with respect to coordinates are used to update structures. We decided to define a pseudo-energy term, differentiate it with respect to the previous predicted coordinates and then provide it to the subsequent block as a $\ell 1$ (vector) feature, the direction of the vectors on atoms in chiral centers breaks the symmetry with respect to reflections in the rest of the input features. The chiral energy function is defined as:

$$E_{chirality} = \frac{1}{N * M} \sum_{i=1}^{N} \sum_{j=1}^{M} (\theta^{i,j}_{\text{predicted}} - \theta^{i,j}_{\text{ideal}})^2$$

where N is the number of chiral centers in the molecule, M is the number of unique pairs of planes that can be constructed starting with a give chiral center that include only explicitly modeled atoms, and $\theta^{i,j}_{\text{predicted}}$ and $\theta^{i,j}_{\text{ideal}}$ are the predicted and ideal angles in chiral center i and set of planes j.

The gradients of the chiral energy with respect to the coordinates are calculated using autodifferentiation in Pytorch [95].

$$chiral\_grads = \text{autograd}(E_{\text{chirality}}, xyz)$$

Constructing *ℓ1* features As shown with the chiral gradients, we can provide vector features for different inductive biases through *ℓ1* features. Inputs of the geometry of the previous N-Cα-C (or OP1-P-OP2) frames are necessary to make accurate updates of frame orientations. We provide these features as $\ell 1$ features representing the vectors between the N and C coordinates with respect to the Cα. These features are also provided for atom nodes with respect to their canonical frame (nodes without frames such as metal ions receive vectors full of 0s for this feature).

Constructing Edge features In the SE(3) Transformer, edge features are used to update nodes. Edges are directed so edge features $e_{i,j}$ are different than $e_{j,i}$. We begin by computing the sequence relative encoding and the atom distance features described in Algorithm 7. We then compute the radial basis function of the pairwise Cα (or P or atom) distances from the previous predicted structure. The *pair* features, the sequence and atom separation features and the radial basis function features are concatenated and fed into a small network. This network consists of a Linear layers and a FeedForward Layer followed by a LayerNorm (same architecture as the network that process the $\ell 0$ *features*. The graph is constructed, with all nodes connected to all nodes (in both directions). The resulting graph, ℓ node features and edge features are processed by the SE(3) Transformer and predicts node-wise $\ell 0$ and $\ell 1$ features.

Applying Structure Updates The output *ℓ0* features are the updated *state* features. The predicted *ℓ1* features correspond to coordinate updates (in nanometers) and four values that are used to generate a quaternion as in AF2. The quaternion is converted into a rotation

matrix and used to update the orientation of N-Cα-C and OP1-P-OP2 frames (the predicted quaternion values have no semantic meaning for atom nodes and the rotation matrices are explicitly set to the identity matrix so gradients are not computed). The first row of the *msa* representation and the *state* features are concatenated and fed into a small residual network and predict χ angles for protein residues and nucleic acid bases as in RF2 and RF2NA.

Refinement Layers

The refinement layers are additional 3D update blocks that do not feed back into the main 1D and 2D updates. In RFAA there are 4 blocks with shared weights, which refine the final structure based on the 1D and 2D features. We make some slight changes from the 3D update in the main blocks. First, there is an additional $\ell 1$ feature. We compute an approximation of the Rosetta Leonard Jones potential (described in subsection 2.10.5) and provide the network the gradient of the potential function with respect to each protein and nucleic acid frame atoms (and the singular atom for each atom node, it does not make sense to use the canonical frame in this case because each node only updates one atom). This is intended to provide the inductive bias to the network that atoms and protein residues should not be clashing. We also provide an extra $\ell 0$ feature which is the gradient of the LJ potential with respect to the χ angle predictions of the network at the previous block. Second, the graph is no longer fully connected. We expect that this stage of the network, the updates are reflective of local interactions and not global movements that would need a fully connected update. The changes are highlighted in Algorithm 10.

Auxiliary Binding Head

The auxiliary binding head is a classifier network that classifies if two chains will bind or not. We trained the network to predict binding for protein complexes and protein-nucleic acid complexes as previously described. We expect future work could use these parameters to learn the task of discriminating binding and non-binding molecules but this was not trained for the RFAA model described in the rest of this work.

Algorithm 10 RFAA Refinement Layer Update

function REFINEMENT_LAYER($msa_{cluster}, pair, state, xyz, idx, bond\_feats, dist\_matrix, chiral\_grads$, clash_l1_grads, $atom\_frames$, clash_l0_grads, $topk$)
  seq = LayerNorm($msa_{cluster0}$) ▷ Query sequence latent embedding
  $pair, state$ = LayerNorm($pair$), LayerNorm($state$)
  $node\_feats$ = Linear(CONCAT([$seq, state$]))
  $node\_feats$ += FeedForward($node\_feats$)
  $node\_feats$ = LayerNorm($node\_feats$)
  $node\_feats$ = CONCAT([$node\_feats, clash\_l0\_grads$])
  rel_pos, bond_dist = Positional_Encoding(idx, bond_feats, dist_matrix)
  rbf_feat = rbf($||xyz_{C\alpha} - xyz_{C\alpha}||$) ▷ Or P or atom coordinate
  $edge\_feats$ = Linear(CONCAT([$pair$, rbf_feat, rel_pos, bond_dist]))
  $edge\_feats$ += FeedForward($edge\_feats$)
  $edge\_feats$ = LayerNorm($edge\_feats$)
  $frame_{ij}, frame_{kj} = frame_i - frame_j, frame_k - frame_j$ ▷ Construct frames from N-Cα-C, O-P-O or atom_frames
  l1_feats = CONCAT([$frame_{ij}, frame_{kj}, chiral\_grads$, clash_l1_grads]
  geometric_feats = $xyz_i - xyz_j$
  $state, coord\_update$ = SE3_Transformer(node_feats, edge_feats, l1_feats, geometric_feats, top_k= topk)
  $quaternion\_values, T = coord\_updates$
  R = construct_rotation_matrix_from_quaternion($quaternion\_values$) ▷ Set R = Identity matrix for "atom" nodes
  $xyz = R \circ xyz + T$
  $sc\_torsions$ = SC_pred($msa_{cluster0}, state$)
  return state, xyz, sc_torsions
end function

Auxiliary Error Prediction

Similar to the original RF, we predict the allatom LDDT of each residue and assess a cross entropy loss on the predictions (over 50 evenly spaced bins). We extend this to atoms by predicting the LDDT of the atom with respect to the other nodes present. We also predict two pairwise accuracy estimation metrics, predicted aligned error (pAE) and predicted distance error (pDE). Predicted aligned error (similar to AF2), aligns each frame (i) and computes errors over all Cα (or P or atom) coordinates. For atom nodes, the canonical frames are aligned and the error is computed over all other nodes. We also developed pDE as an alternative which just predicted the unsigned distance error between any two nodes. We found empirically that pAE correlates better with accuracy so we do not report pDE statistics.

2.10.5 Loss Functions

Resolving Equivalent Atom Orderings

When predicting general biomolecular systems with systems at hybrid residue and atom resolutions, there are multiple symmetric atom labels that represent the same biomolecular system. First, certain protein sidechains contain 180° flips. Second, certain assemblies contain multiple identical protein chains which could be predicted in any order. Third, arbitrary assemblies might have multiple copies of the same small molecules that could be predicted in any order. Fourth, arbitrary small molecules can have symmetric chemical groups that could be reordered. We developed an algorithm which assigns the ordering that would minimize distance RMSD between a ground truth assembly and a predicted assembly. We handle symmetric swaps of sidechains similarly to AF2 and RF2, where the distances of each ambiguous atom to all non-ambiguous atoms is computed for the predicted and true structure and the closest atom naming assignment to the true structure for each permutation swap is assigned. The process for resolving the remaining symmetries is as follows: 1) Distances between Cα coordinates in the predicted structure and Cα distances in all valid permutation swaps of protein chains (see subsection 2.10.3) are computed. Chains are assigned by finding the chain ordering that has the minimum difference between the distances in the predicted structure and distances in the true structure. 2) Given the anchor point of the protein ordering, every ligand is greedily assigned by the minimizing the difference between distances between atoms in the ligand to Cα atoms in the predicted protein structure and true protein structure (based on the assignment in step 1). During the greedy assignment, all possible permutation swaps of atoms within each ligand are also considered.

Masked Token Recovery

The masked token recovery loss is identical to the published RF2 [11, 39, 103]. In cases with small molecules or other biomolecules, we only mask protein tokens so there is no masked token recovery loss applied for other biomolecules.

Torsion Angle Loss

The torsion angle loss is unchanged from RF2 and RF2NA. Atom nodes have no torsion angle loss applied, since none of those values have semantic meaning with respect to placing individual atoms.

Distogram Loss

Following trRosetta [140] and RoseTTAFold [10], we apply a loss on predicted pairwise distance and orientations from the final pair representation. Several modifications were made to accomodate arbitrary biomolecules. First, we changed the binning of the distograms. For proteins in AF2 and RF2, the distogram loss is evenly spaced from roughly 2.5Å to 20Å. For small molecules, shorter distances (bond lengths, hydrogen bonds) are often more important so there are 30 evenly spaced distogram bins between 1Å and 4Å and then 30 more (coarser) bins between 4Å and 20Å. Three more angle losses representing inter-residue orientations are applied to a projection from the *pair* features. For atom nodes, the canonical frame is used to compute the three points for each angle. First, we compute a pseudo Cβ atom for every frame (using default Rosetta params):

$$\vec{x} = b - a$$

$$\vec{y} = c - b$$

$$\vec{z} = \vec{x} \times \vec{y}$$

$$d = -0.57910144\vec{z} + 0.5689693\vec{x} - 0.5441217\vec{y} + b$$

where a, b, c are three atom coordinates in a frame. While this coordinate (d) does not mean anything physically for nucleic acids or atom frames, we reasoned that this was a consistent geometric quantity so the network should be able to predict quantities derived from it given a local coordinate frame.

$D_{:,l,l'}$, $\Omega_{:,l,l'}$, $\Phi_{:,l,l'}$, $\Theta_{:,l,l'}$, together describe the orientation of residue l relative to residue l'. The following loss consists of the cross entropy between the one-hot histogram of the known inter-residue distances and orientations and the corresponding distributions predicted by the model.

$$\begin{aligned}\mathcal{L}_{\text{2D}}(\text{logits}_d, \text{logits}_\omega, \text{logits}_\theta, \text{logits}_\phi, z_0) =& \text{CrossEntropy}(\text{logits}_{\text{dist}}, \mathbf{D}) + \\ & \text{CrossEntropy}(\text{logits}_\omega, \Omega) + \\ & \text{CrossEntropy}(\text{logits}_\theta, \Theta) + \\ & \text{CrossEntropy}(\text{logits}_\phi, \Phi)\end{aligned}$$

where:

$$\begin{aligned}&D \in \mathbf{R}^{[\mathbf{C_{dist}xLxL}]}; D_{b,l,l'} = \mathbb{1}[\text{bin}_{D,b}^{low} \leq \max(\|C_{\beta,l'} - C_{\beta,l'}\|_2, 18.5) < \text{bin}_{D,b}^{high}] \\ &\Omega \in \mathbf{R}^{[\mathbf{C_{dist}xLxL}]}; \Omega_{b,l,l'} = \mathbb{1}[\text{bin}_{\Omega,b}^{low} \leq \text{Dihedral}(C_{\alpha,l}, C_{\beta,l}, C_{\alpha,l'}, C_{\beta,l'}) < \text{bin}_{\Omega,b}^{high}] \\ &\Theta \in \mathbf{R}^{[\mathbf{C_{dist}xLxL}]}; \Omega_{b,l,l'} = \mathbb{1}[\text{bin}_{\Theta,b}^{low} \leq \text{Dihedral}(N_{\alpha,l}, C_{\alpha,l}, C_{\beta,l}, C_{\beta,l'}) < \text{bin}_{\Theta,b}^{high}] \\ &\Phi \in \mathbf{R}^{[\mathbf{C_{phi}xLxL}]}; \Omega_{b,l,l'} = \mathbb{1}[\text{bin}_{\Phi,b}^{low} \leq \text{Planar}(C_{\alpha,l}, C_{\beta,l}, C_{\beta,l'}) < \text{bin}_{\Phi,b}^{high}]\end{aligned}$$

and the bin edges for converting these angles and distances into a one-hot distribution are given by:

$$\text{bin}_{D,i} = \left\{ \begin{array}{ll} [0, d_{min}], & \text{if } i = 0 \\ [d_{min} + (d_{mid} - d_{min})\frac{i-1}{29}, d_{min} + (d_{mid} - d_{min})\frac{i}{29}], & \text{if } 1 < i < 30 \\ [d_{mid} + (d_{max} - d_{mid})\frac{i-30}{30}, d_{mid} + (d_{max} - d_{mid})\frac{i}{30}], & \text{if } 30 =< i < 60 \\ [d_{max}, \infty], & \text{if } i = 60 \end{array} \right\}$$

$$\text{bin}_{\Omega,i} = \text{bin}_{\Theta,i} = [-\pi + \frac{2\pi i}{37}, -\pi + \frac{2\pi(i+1)}{37}]$$

$$\text{bin}_{\phi,i} = [\frac{\pi i}{19}, \frac{\pi(i+1)}{19}]$$

where $d_{min} = 1.2$, $d_{mid} = 4$, and $d_{max} = 20$

The formulas for computation of dihedral and planar angles are given by

$$\begin{aligned}\text{Dihedral}(a,b,c,d) = \text{atan2}(&\\ &[c-b]\cdot(([b-a]\times[c-b])\times([c-b]\times[d-c])),\\ &\|c-b\|([b-a]\times[c-b])\cdot([c-b]\times[d-c]))\\ \text{Planar}(a,b,c) = \arccos(&\frac{(a-b)\cdot(c-b)}{\|a-b\|\|c-b\|}).\end{aligned}$$

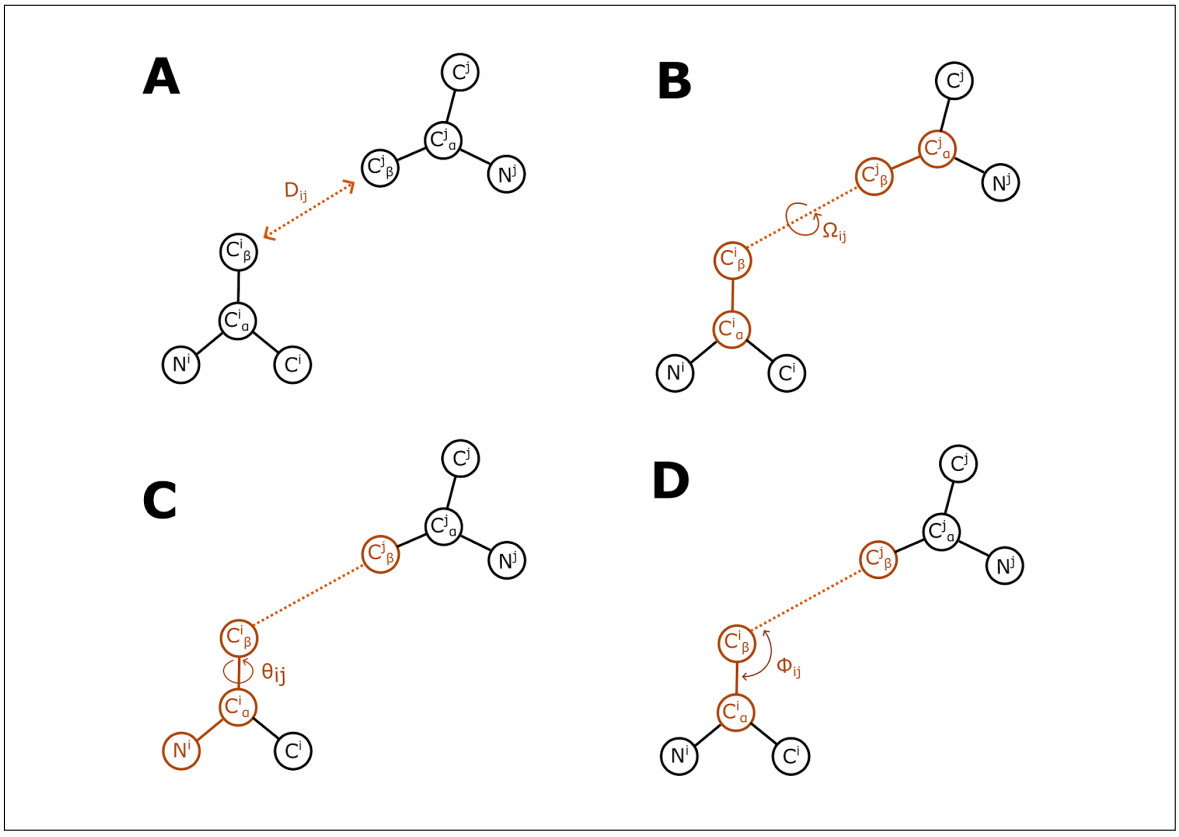


Figure 2.24: Diagrams for how to compute the four inter-residue distance and dihedral degrees of freedom.

All-Atom FAPE

Based on our experience training RF2 and RFNA, the FAPE loss introduced in the AF2 paper [72] was essential to producing accurate structures with the correct chirality. In RF2 and AF2, FAPE was applied only on the backbone frames for the intermediate generated structures and the full atom FAPE was applied on the final step and backpropagated through the whole network. Analogous to using N-Cα-C frames, we reasoned that we could use triplets of bonded atoms as frames, align the predicted and true atoms and compute the error over the other atoms.

The easiest way to do this to enumerate all triplets of bonded neighbors for each atom and apply a loss on the deviations on all atoms when aligning all the potential frames. For the "backbone" frame loss applied across all 40 (4 *msa_full* blocks, 32 main blocks and 4 refinement layers), we wanted to choose a single frame to apply a loss over for each atom node. We decided to implement an algorithm that deterministically chose a *canonical* frame for each atom node as described in subsection 2.10.2. Each atom's *canonical* frame is used for the computation of FAPE. We believe that deterministic frame selection is important because it ensures FAPE is a stable optimization objective as if frames were stochastically sampled, the same predicted structures could have different FAPE values depending on the choice of frames but did not test any alternatives. We also believe that deterministic frame selection is important for the frame orientation and predicted alignment error losses because the network has to predict quantities dependent on the chosen frame.

During early experiments, we found that the network was able to move groups of atoms closer to the desired pocket but did not make coherent ligand structures or interactions with the protein. We reasoned that there were two issues with our formulation of FAPE. First that the scale of distances that are important for small molecule structures are smaller than those that are important for proteins. Second, in a given training example, most nodes are protein residues so the errors of protein residues outweigh the errors of small molecules in the total loss calculation and the gradient to the weights of the network. We added an additional loss term that consisted of only the errors of the atom node coordinates with respect to the

atom node frames with a normalization value of 4Å (compared to 10Å in AF2 and RF2). We also found that increasing the relative contribution of this term to the total loss helped balance the contributions of the protein errors and small molecule errors.

The second phenomenon (less contributions from atom nodes) also affected the contribution of the interchain FAPE values (errors of protein atoms with respect to atom frames and errors of atom coordinates with respect to protein frames). We chose to also add an auxiliary loss consisting of the interchain FAPE values and scaled its relative contribution to the total loss.

In the early training, we use an exponential schedule to weight backbone FAPE over the different layers of the network. During the fine-tuning stage of training, we equally weight FAPE across all layers. The following equation is used to weight backbone FAPE over all the intermediate layers of the network:

$$\mathcal{L}_{\text{fape backbone}} = \frac{1}{\sum_{i=0}^{I-1} \gamma^i} \sum_{i=1}^{I} \gamma^{I-i} \text{FAPE}(x^{(0)}, \hat{x}^{(0),i})$$

where $\hat{x}^{(0),i}$ is the i^{th} structure block output and γ is a scaling factor. The hyperparameters used for the different FAPE terms are shown in Table 2.9. We tuned these values in very early experiments and did not optimize them as we updated the network architecture so it is possible that better values exist for final performance.

To compute frames from an arbitrary set of three points we use a Gram-Schmidt process (identical to the rigid_from_3_points algorithm in AF2; referred to as Gram_Schmidt). Unlike AF2, we use this process on the predicted coordinates as well instead of using the chained set of rotations predicted by the network since small molecule nodes do not predict rotations. To compute FAPE, we first assemble a set of indices that comprise each frame, construct the 3 points for each frame, compute the associated rigid orientation of each frame in the global frame and apply the operation on each the predicted and true coordinates and measure the errors (shown in Algorithm 11).

Algorithm 11 Compute All-Atom FAPE

function COMPUTE_GENERAL_FAPE(pred, true, atom_frames, Z, d_clamp)
    frames = construct_frames(atom_frames) ▷ Predefined frames for proteins/NA, atom_frames for atoms
    $pred_{frames}$ = gather(pred, frames) ▷ N_{nodes}, M_{frames}, 3 atoms,3 coordinates
    $true_{frames}$ = gather(true, frames) ▷ N_{nodes}, M_{frames}, 3 atoms,3 coordinates
    T_{pred}^{-1} = Gram_Schmidt($pred_{frames}$)
    T_{true}^{-1} = Gram_Schmidt($true_{frames}$)
    $x_{pred} = T_{pred}^{-1} \circ pred$
    $x_{true} = T_{true}^{-1} \circ true$
    $e_{ij} = \sqrt{(x_{pred} - x_{true})^2}$
    $\mathcal{L}_{fape} = \frac{1}{Z} mean_{ij}(clamp(e_{ij}, d\_clamp))$
    return $\mathcal{L}_{fape}$
end function

Table 2.9: Parameters for different FAPE terms

Term	d_clamp	Z
$\mathcal{L}_{fapefullatom}$	10	10
$\mathcal{L}_{fapebackbone}$	90% 30, 10% unclamped	10
$\mathcal{L}_{fapeintra-ligand}$	4	4
$\mathcal{L}_{fapeinterproteinligand}$	90% 30, 10% unclamped	10

Bond Geometry Losses

Generally, we found that the network learned accurate bond lengths and angles from the training set. We did notice that in some cases when the network was unsure on docking, it would produce unideal bond lengths and angles. We decided to place a small loss on bond lengths, angles, and distances within rigid groups (rings). We use a smoothed L1 Loss with beta=0.2. For bond lengths and rigid groups, we apply the loss on all pairwise distances (between two bonded atoms or atoms in a shared rigid group). For angles, we place the loss on distances predicted between every atom and atoms 2 bonds away in the bond graph. We reason that this in conjunction with the loss on bond lengths is sufficient to constrain the angles as well (similar ideas to those presented in [118]). We also apply this loss on bond lengths between atomized N atoms and the previous residue C and atomized C atoms and the following N. Losses on backbone geometry for proteins and nucleic acids are exactly the same as those in RF2. Bond lengths and angles are penalized compared their ideal values with a tolerance and the errors are normalized by the amount of violating predictions.

Clash Loss

We found that initial trained versions of the network produced some clashes between protein residues and between protein residues and small molecule atoms. Following RF2, we intended to use a modified version of the Rosetta Leonard Jones potential (6-12 potential) [7] as an additional loss term to penalize the network from making clashing structures. Rosetta uses atom typing to determine the optimal parameters for the LJ potential and since our network operates on a larger set of atoms than have been tuned by the developers of Rosetta, we generated a set of parameters to use in our loss. These roughly correspond to the the mode of the parameters for each element type (most common parameter across all atom types for each element), parameters from UFF [104] where there are no available Rosetta parameters and default values for when there are no UFF or Rosetta parameters. We did not tune these values during model development.

Table 2.10: LJ Loss Parameters for Atom Tokens

Element Type	LJ Radius	LJ Well Depth
Al	1	0.1
As	1	0.1
Au	1	0.1
B	1	0.1
Be	1	0.1
Br	2.1971	0.1090
C	2.0067	0.0689
Ca	1	0.1
Cl	2.0496	0.1070
Co	1	0.1
Cr	1	0.1
Cu	1	0.1
F	1.6491	0.0750
Fe	1	0.1
Hg	1	0.1
I	2.36	.111
Ir	1	0.1
K	1	0.1
Li	1	0.1
Mg	1	0.1
Mn	1	0.1
Mo	1	0.1
N	1.7854	0.1497
Ni	1	0.1
O	1.5492	0.1576
Os	1	0.1
P	2.1290	0.5838
Pb	1	0.1
Pd	1	0.1
Pr	1	0.1
Pt	1	0.1
Re	1	0.1
Rh	1	0.1
Ru	1	0.1
S	1.9893	0.3634
Sb	1	0.1
Se	1	0.1
Si	1	0.1
Sn	1	0.1
Tb	1	0.1
Te	1	0.1
U	1	0.1
W	1	0.1
V	1	0.1
Y	1	0.1
Zn	1	0.1

2.10.6 Training Details for RFAA Structure Prediction

RFAA was trained in three phases which were intended to interrogate the question of whether a single model could represent all biomolecules. We iteratively added in different datatypes and tracked accuracy using the held out validation clusters. We observed that adding new datasets did not decrease accuracy on datasets that were already present in training. We anticipate that the network could be trained on all the datasets simultaneously and achieve the same accuracy but have not tested this rigorously. The first two phases were the main training phases and the final phase was a fine-tuning phase with different hyperparameters and loss functions.

Training Hyperparameters

The first two stages of training are analagous to the initial training of RF2 and AF2 and the third stage is similar to the "fine-tuning" stage where violation losses are turned on, crop size increased and hyperparameters changed to refine the accuracy of the network. Exact hyperparameters used are shown in Table 2.11.

Dataset Sampling

At the outset of this work, it was unclear to us whether it would be possible to train a single network to model all biomolecules in the PDB. We decided to add in new datasets as training progressed to assess whether the network could simultaneously learn features from all datasets. We restarted training in three phases with different dataset proportions. We found generally adding new datasets did not decrease accuracy on the datasets that were already trained, although it is unclear whether we would have seen further improvements if we continued training without adding the new datasets.

Each dataset was sampled with a given probability and then each sequence cluster within that dataset is sampled with inverse probability to the number of examples in the cluster. Within the protein monomer clusters, if examples are homomers we stochastically sample featurizing them as monomers or homomers ($p_{\text{homomer features}} = 0.5$).

Table 2.11: Training Hyperparameters

	Phase 1	Phase 2	Phase 3
Crop Size	256	256	375
Batch Size	64	64	128
Learning Rate	0.001	0.001	0.0002
Schedule	No warmup. Decay Learning Rate by 0.95 every 5000 steps	No warmup. Decay Learning Rate by 0.95 every 5000 steps	No warmup. Decay Learning Rate by 0.95 every 5000 steps
Loss	$10 * \mathcal{L}_{fapefullatom} + 5 * \mathcal{L}_{fapebackbone} + 10 * \mathcal{L}_{fapeintra-ligand} + 2.5 * \mathcal{L}_{fapeinterproteinligand} + 1.0 * \mathcal{L}_{c6d} + 3.0 * \mathcal{L}_{msamask} + 10 * \mathcal{L}_{torsion} + 0.1 * \mathcal{L}_{plddt} + 0.5 * \mathcal{L}_{pae} + 0.5 * \mathcal{L}_{pde}$	$10 * \mathcal{L}_{fapefullatom} + 5 * \mathcal{L}_{fapebackbone} + 10 * \mathcal{L}_{fapeintra-ligand} + 2.5 * \mathcal{L}_{fapeinterproteinligand} + 1.0 * \mathcal{L}_{c6d} + 3.0 * \mathcal{L}_{msamask} + +10 * \mathcal{L}_{torsion} + 0.1 * \mathcal{L}_{plddt} + 0.5 * \mathcal{L}_{pae} + 0.5 * \mathcal{L}_{pde}$	$10 * \mathcal{L}_{fapefullatom} + 5 * \mathcal{L}_{fapebackbone} + 10 * \mathcal{L}_{fapeintra-ligand} + 2.5 * \mathcal{L}_{fapeinterproteinligand} + 1.0 * \mathcal{L}_{c6d} + 3.0 * \mathcal{L}_{msamask} + 10 * \mathcal{L}_{torsion} + 0.1 * \mathcal{L}_{plddt} + 0.5 * \mathcal{L}_{pae} + 0.5 * \mathcal{L}_{pde} + 0.5 * \mathcal{L}_{bind} + 0.02 * \mathcal{L}_{protein,NAgeom} + 0.02 * \mathcal{L}_{atomgeom} + 0.02 * \mathcal{L}_{clash}$
top_k in Refinement Layers	128	128	64
Exponential Decay of FAPE over Str Layers	0.99	0.99	1.0
Optimizer Steps	~35e3	~40e3	~15e3

Table 2.12: Dataset Sampling Proportions

Dataset	**Phase 1**	**Phase 2**	**Phase 3**
Protein Monomer	0.09	0.07	0.12
AF2 Distillation	0.0	0.14	0.36
Protein Complex	0.17	0.11	0.055
Negative PPI	0.0	0.0	0.055
Protein-Nucleic Acid Complex	0.17	0.11	0.055
Negative Protein-Nucleic Acid Complex	0.0	0.0	0.055
RNA	0.09	0.06	0.05
Protein-Small Molecule Complex	0.37	0.20	0.11
Protein-Metal Complex	0.0	0.10	0.03
Protein-Multi-residue Molecule Complex	0.0	0.05	0.0275
Covalently Modified Protein	0.0	0.05	0.0275
Protein-Small Molecule Assembly	0.0	0.0	0.055
CSD Crystal Structures	0.03	0.03	0.03
Atomized Protein Augmentation	0.08	0.08	0.04

We do not believe that the exact values of the dataset samples are essential to recover the accuracy that we observe in the paper. The values were tuned with three guiding principles, 1) to balance out sequence clusters sampled early in training, 2) to show equal amount of positive and negative examples and 3) to bias towards the larger datasets towards the end of training to avoid overfitting on smaller datasets. The dataset sampling proportions are shown in Table 2.12.

2.10.7 Structure Prediction Inference Regimen

At inference time, we run the model without MSA corruption or template subsampling, instead preserving all MSA tokens and picking the top 4 searched templates for each entry. We run the model for 10 recycles unless otherwise specified. For every protein sequence, we build MSAs and templates using the standard MSA and template generation pipeline described in subsection 2.10.1.

We define the *ligand RMSD* as follows: for each ligand target, we compute in the crystal structure every backbone atom that is 10Å of the any atom in the bound ligand. We then kabsch align the predicted and crystal protein structures on the aforementioned backbone atoms, and use the same transformation matrix to superimpose predicted and crystal pose ligands. The ligand RMSD is the resultant RMSD between predicted and true ligand positions. We use this metric throughout our evaluation of protein-ligand complex predictions. We note in figure legends when ligand RMSD is computed by an external tool (eg. for CAMEO evaluations and Posebusters).

The model makes predictions of its own error during training time in a 2D matrix called the predicted alignment error as desrcibed in 2.10.4. The i, jth entry is trained to be an estimate of the ℓ^2 distance error between the the jth atom in the ith coordinate frame. We define the inter-chain protein-ligand predicted alignment error (PAE Interaction) as the mean of the predicted alignment error tensor between all protein residue frames and small molecule coordinates, and all small molecule frames and residue coordinates.

We do not perform any cropping at inference time, predicting full complexes (including residues/atoms that were not resolved in the true structure).

CASP14 Protein Monomer Targets

We benchmarked the performance of RFAA against RF2 and AF2 using a subset of the CASP14 targets with experimentally resolved structures that were not removed during competition. TBM-easy targets were discarded and the final set of 42 proteins was composed of monomeric targets from TBM-hard, FM/TBM, and FM categories. Each of the three

methods were run using default parameters with the same input multiple sequence alignments and no templates.

- AF2 (V1 weights), model_1, and 20 recycles
- RF2 (Apr23 weights), model 1, and 20 recycles
- RFAA (this work), model 0, and 20 recycles

GDT-TS was calculated based on experimentally solved structures provided by CASP using TM-score (accessed Jun, 2023) [147]. MSAs were generated as described in [9]. Briefly, query sequences were searched iteratively with HHblits [108] against uniclust30 (UniRef30_2020_01) database [1] with a gradient of E-value cutoffs and 95% sequence identity filtering. For targets with shallow MSAs, we converted the uniclust30 generated MSA into a seed HMM to search against JGI [30] with hmmsearch [43]. Homologous sequences from this search were aligned and combined with previous uniclust30 sequences.

CAMEO Targets

The CAMEO BETA challenge (`https://beta.cameo3d.org/`) is an online, weekly, continuous evaluation of the most recent depositions into the PDB. We registered the RFAA model as a server for homomeric and ligand targets (excluding RNA, DNA and heteromeric targets for simplicity). The RFAA server searches for MSAs and templates for each protein sequence as described above. The server makes 5 different predictions with different random seeds for each target and then submits the prediction with the lowest interchain PAE. If there are multiple ligands in a given target, the interchain PAE is calculated over all ligands in the target.

We ran the RFAA CAMEO server (Server 2) from 04/08/2023 to 09/02/2023. Ligand pose baselines were introduced from 08/12/2023 onward (`https://beta.cameo3d.org/cameong_servers/`). The CAMEO organizers returns ligand RMSD poses scores that we then plot for our evaluations. We do not compute our own RMSD metrics for the CAMEO challenge.

The CAMEO server does not provide stoichiometry information for either protein or ligand in a given protein ligand complex. In order to determine protein stoichiometry, we check the symmetry group of each template of the input protein. We make a list of the top 20 most similar templates to the input protein by sequence similarity, and then filter that list based on a coverage threshold of 0.75 and a match score of 0.95. For the passing templates, if at least half of them form a dimeric or a trimeric complex, we then attempt to predict a dimeric or trimeric complex with the input protein, respectively. If the predicted alignment error between the symmetric subunits is less than 10 and the computed clash score between the subunits is less than 1, we model the protein as either a dimer or a trimer. Otherwise, we default to modeling the protein as monomeric.

We only make predictions with a single copy of each ligand in each target and do not duplicate ligands in the case of dimer or trimer-forming protein chains. We found it difficult to distinguish between cases where there are two binding pockets for two different copies of the ligand in a dimer compared to a single copy of the ligand sitting at the dimeric interface. We also do not model symmetry groups above C3 because they often do not fit in GPU memory on the allocated GPUs.

Recent PDB Evaluations

Protein Nucleic Acid Complexes We evaluate RFAA on Protein Nucleic Acid complexes using a dataset of recently deposited PDBs curated in [12]. RFAA and RFNA are evaluated using the same MSAs and templates using the default parameters described. For this benchmark, all small molecule and noncanonical amino acid/base context is ignored.

We repredicted a small subset of predictions where the protein-NA interface was predicted accurately and there was small molecule also bound near the NA interface (an example shown in Fig 2D). We did not rigorously test the model's ability to model ternary complexes and expect that adding them into training (perhaps by fine-tuning) would significantly improve accuracy.

Small Molecule Protein Complexes We curate a list of protein-ligand complexes deposited in the PDB from 2021 and onwards. For each ligand in every entry in the PDB past this deposition date, we make a dataset of items defined by the procedure outlined in 2.10.1. We add the additional constraint that every ligand in this evaluation set must be designated as a “subject of investigation", a label which indicates that the authors who deposited the crystal structure believe the ligand to have some significance rather than being, for example, a solvent molecule. We note that this is not a perfect filter: certain PDB entries have ligands marked as subject of evaluation that are not referenced in the main text of the associated papers.

In order to remove redundancy from this evaluation set, we clustered the primary protein partners of each item down to 30% sequence identity at 80% coverage using MMSeqs2 [116]. For each sequence cluster, we compute the set of unique query ligands in that cluster and select, uniformly at random, a single protein-ligand complex for each unique query ligand. This clustering process ensures that we do not bias our evaluations by having many repeated copies of similar protein ligand binding pockets. For each item in the dataset, which consists of a query ligand and its immediate protein and non-polymer binding partners, we compute the total length (number of residues + number of atoms) of the assembly and then filter out all items of length > 1000.

The final held-out evaluation dataset consists of 5421 items. Of these, 897 consist of ligands that are covalently bonded to the protein and 622 of them are metal ions.

PoseBusters The PoseBusters dataset carefully curated subset of the protein-ligand complexes in the PDB released on or after 2021 designed for evaluating the performance of docking methods [25]. Each item in the PoseBusters dataset consists of a PDB entry and a ligand identifier specifying the ligand to be docked. There are 428 protein-ligand complexes with non-redundant protein chains and non-redundant, drug-like ligands.

We pre-process the data in largely the same way as we described in subsection 2.10.1, with a few distinct changes. First, the posebusters dataset pre-specifies the *query ligand* of interest.

Second, we narrow the definition of contacting protein chain to be any protein that has at least one atom within 10Å of the query ligand. Third, we define any contacting cofactors to the query ligand as any cofactor that has at least one atom within 5Å of the query ligand. Finally, if there are multiple identical cofactors within the context of the query ligand, we keep only one such copy. This slightly narrowed definition of query ligand context allows the network to focus more on docking the query ligand, which is the purpose of the PoseBusters evaluation.

For each item in the PoseBusters evaluation set, we predict the entire complex - query ligand, contacting protein partners, and contacting cofactors - simultaneously. We superimpose the primary protein partner onto the crystal structure and extract the crystal pose of the query ligand. We then run the posebusters suite on the predicted pose of the molecule to determine chemically validity of the predicted ligand pose. We also use the posebusters suite to compute ligand RMSD between crystal and predicted ligand pose after protein superimposition, and to compute validity metrics between predicted ligand pose and *predicted* protein structure (and predicted cofactors).

We note that our model is not a docking method and does not take as input the crystal structure of the protein nor the binding site of the query ligand, making our task (prediction from sequence information alone) strictly harder than blind docking or local pose estimation. This implies that it does not make sense to compute validity metrics, such as minumum distance clashes, between predicted ligand pose and *crystal* protein structure, which is why we use the model's predicted protein structure instead.

Finally, we also note that although our model is only trained on data from the PDB up to April 2020 and the PoseBusters dataset is only defined on data from 2021 onwards, our training data is defined on the deposition date of the PDB entries, while the PoseBusters dataset is defined on the entry release date. In almost all cases the dates of deposition and release are similar enough that the items in the PoseBusters evaluation set are not in our training set. However, a single entry (6VTA, AKN) was deposited before April 2020 but released in 2021. We exclude this item from the evaluation set for all methods.

Covalent Modifications Our evaluation set of covalent modification consists of the 897 entries described in subsection 2.10.7. These items underwent the same filtering described in subsection 2.10.1. We excluded all metal ions that have *covale* headers in the PDB because these are trivial modeling cases. During evaluations, we split the predictions into three categories based on their putative biological functions shown in Table 2.13. We used the PDB 3-letter codes to identify cases from our evaluations in each set. For multi-residue ligands with covalent bonds, we included them in a subsection if any of the residues were present in any of the categories. There were no multi-residue ligands where multiple residues were in multiple categories.

Table 2.13: Types of Covalent Modifications

Type of Modification	PDB 3-letter Codes
Glycosylation	BGC,MAN,NAG,FUC,BMA,GLC,RIB NGC,GAL,BDP,A2G,JIW
Enzyme Cofactors	PLP,HEC,F3S,SAH,ADP,ATP,UDP, FAD,AMP,FMN,ANP,HEM

Other	UVT, 24N, V4B, 2GI, 9JT, 2I5, 21I, V48, TJB, V3Q, 1XZ, WLP, TJ8, 2I8, V4N, TKK, SGM, HC4, S7Q, ZFG, S7N, RET, 9IX, N36, V0G, S08, DWZ, UZM, 2RG, WV0, S0Q, UQN, MW0, UZS, RZZ, PNS, IM2, RY2, CYC, ID1, 2LJ, ISS, UZV, RET, 4D6, 30W, FDX, N2Q, NXL, RXW, RET, 1S6, CMC, QS8, ZZ7, UPQ, MER, MWC, UZY, L9U, HH8, L8X, IIY, IJI, IK3, HNU, 82Q, LW1, L9C, 8NB, LBI, V7G, VR4, V46, YHI, XTM, YHJ, 2RG, UHS, UJ1, DWZ, XV4, VEV, 9JT, Q8H, PLM, USA, VEG, VEJ, XTP, RW8, XTJ, 0WN, CHL, VEM, UHY, DWZ, ZL7, UPD, VEP, 9JT, 2RG, ZJ1, Q8E, USD, UHV, VEY, YCV, UPJ, XC4, W48, UED, Y48, UJ4, EKZ, UUK, O1K, US8, 9ZG, 2IE, USZ, 2IJ, QVR, UQW, YWJ, 5YZ, URK, AMI, USH, SUU, R1L, UED, O1R, UUW, 1LE, 5ZB, ALD, A0U, O10, RN2, B1S, 7VB, QNC, AG7, 4W8, IFO, 91Z, NNA, SIN, U5G, 8GW, 7YW, I8H, SV6, PXQ, SFW, 90X, I1W, 8T6, I68, I71, 7VQ, 7W5, 4WI, 90U, R8H, 7YB, E9H, RBL, IRR, 7YZ, Q56, 8ZI, H60, H63, MYC, I70, EOF, 4IT, 2XI, HC1, IS5, 99W, TG3, H6L, 800, 7VW, 90I, HF0, I80, 7XK, C7A, HER, IRZ, 7VI, I54, NEN, 90H, G7L, 9SW, CB1, 8UL, K36, 2RI, ALD,

Other (continued)	Z41, 2DA, FEY, HYR, FHS, FZI, ACM, FVE, 1RG, BOV, U88, 1S7, R28, WFD, M1V, PNS, 90F, VOY, HHL, VO7, WF7, 5JR, NXL, 70I, BVX, AR6, RQT, RQZ, DTT, R1W, NFF, R2E, WZG, ZN, UOT, P5N, UPK, MHH, UON, 8I4, VX5, UFH, TJ8, RET, P8K, TCK, PLM, VLD, RET, X2S, MU4, YS7, X2G, DUT, YY3, DYF, X2J, PLR, Y37, 8BS, C9D, 4KZ, THR, I8K, S1N, MUB, RET, U8W, UT8, RET, 7J8, FP6, RET, DPM, U8Z, USW, RET, UST, NZ6, H40, PXQ, RFT, J2C, 8Z6, FAR, J3X, 88T, JRA, H2T, H0O, I6T, VU1, QTU, VU4, J50, V4T, ACE, ZGV, VLE, NZX, I7H, FHR, W6X, QPE, MYR, BZ2, G7F, PLM, QTU, QPB, Q5W, G7L, CLR, MTX, MEZ, CA, HNO, YJG, G5U, PLM, G8C, BFB, OCA, DON, UVB, H2S, V3N, V48, NEN, UWK, V2E, UVH, V2W, V3W, V32, V1H, V3Z, UXN, V3K, V2N, UVN, CPS, V3T, UWQ, V0T, V1E, V42, V2T, 4Y8, V3B, V1Q, VEE, RVW, UWH, V4K, V2K, V4H, UUB, V1K, V2Q, V0W, PAM, 9S5, 9TC, 7ID, CYC, 9Y6, NAP, S1K, S1B, M1V, VL5, CR8, PEB, ZXQ, DBV, VMM, PMS, WHL, FFQ, M1V, 9SS, GIT, S0Z, CYC, S1E, CYC, SQW, Q5O, 7TC, NW3, CEF, TSL, J00, ZXQ, Q5X, PLR,

Other (continued)	PLM, D12, W6X, SPH, RET, RET, H8S, H8V, T5N, V0Q, PNS, QE8, T6H, T4Z, D10, T6W, UHZ, T4W, OEH, T4Q, 868

Metal Ions 622 of the entries in subsection 2.10.7 are metal ions. As in our training data, we filter out examples that are not in Table 2.2.

Computing Sequence Similarity to Training Set

For each item in our evaluation set, we BLAST [26] (with the `-qcov_hsp_perc 50` option and otherwise default parameters) the primary protein partner against all protein chains seen in our training set and compute the maximum sequence identity percentage to any protein chain seen during training. We consider a validation protein to have a training set match if it there exist a protein seen during training that has $> 30\%$ sequence identity at 50% coverage.

Computing Ligand Similarity to Training Set

To compute similarity of ligands, we download idealized coordinates of all ligands in the PDB in sdf format (see `https://www.rcsb.org/downloads/ligands`). We use OpenBabel to compute the Morgan fingerprint for each ligand. We compute, for every ligand in the evaluation set, the bitwise similarity between the Morgan fingerprints for that ligand and every ligands deposited in the PDB, which we henceforth refer to as the Tanimoto similarity. We consider a ligand "similar" to a ligand seen in training if it has a Tanimoto similarity $>$ 0.5 to any ligand seen during training.

Evaluating effects of protein similarity and ligand similarity to training set on accuracy

In order to investigate whether or not RFAA can generalize to examples beyond those seen during training, we consider two notions of "similarity" to the training set: protein sequence similarity and the aforementioned ligand similarity.

To evaluate the dependence of protein similarity on accuracy, we clustered the sequences curated in subsection 2.10.7 using MMSeqs2 (since we took every unique protein-ligand pair, certain protein sequence clusters were overrepresented). To determine whether it was possible for the network to produce accurate predictions with that sequence cluster, we extracted the lowest ligand RMSD prediction in each cluster. We then BLAST all the cluster representatives against the training set to determine the maximum similarity to any training example.

We followed a similar procedure to measure the impact of ligand similarity on accuracy. We computed an all by all matrix of Tanimoto similarity in our test set. We remove multi-residue ligands from this evaluation since the PDB does not provide ideal coordinates with the entire bonded ligand (only each residue) and the construction of the files can affect the Morgan fingerprint calculation. We then performed agglomerative clustering [96] with a threshold of 0.5 and selected a random representative from each cluster. All of the PDB 3-letter codes from the cluster representatives are mapped back against all the 3-letter codes used during training to measure the maximum Tanimoto similarity between each test set item and each training set item.

Evaluating Correlation between RFAA Accuracy and Native Complex energies

We expected that a network that has learned general principles about protein-small molecule binding would make more accurate predictions for tighter binding interfaces. We performed Rosetta energy calculations on the native complexes from our test set (described in 2.10.7). Since small bond length and angle deviations in native structures can cause large energy differences, we choose to run a short minimization protocol to equilibrate the structures into the Rosetta forcefield. We use a recent protocol described in [94] where Rosetta's small molecule docking method, GALigandDock, is used in "eval" mode to evaluate a pose. In this mode, the ligand and any residues within a heavy atom contact distance of less than 8Å can move. The energy of the conformation is minimized using GALigandDock's generalized ligand potential and a harmonic constraint on the starting coordinates to prevent large movement. The pose is then scored using the generalized ligand potential in Rosetta. We

remove cases where Rosetta does not have appropriate atom-typing or where the ligand moves more than 0.5Å from the starting position since those do not accurately score the native binding complex.

2.10.8 Training Details for RFdiffusionAA

RFDiffusionAA was trained on protein monomer structures in the PDB used for RFAA training 50% of the time and protein monomer/small-molecule complexes 50% of the time. Two small changes were made to the training set for small molecules: first, all cases with small molecules with unresolved atoms were removed and second, chains longer than 384 residues were removed (following RFDiffusion). As in RFDiffusion, training examples consist of the unconditional task 20% of the time and a motif-conditional task 80% of the time. The motif-conditional task comprises three distinct tasks:

- Middle motif [40%]: A contiguous set of residues
- Terminal motif [40%]: Two contiguous sets of residues at the N and C terminus of the monomer.
- Sparse contacts [20%]:
  - *For proteins*: A random set of 3 residues all > 10 residues apart in sequence space but with pairwise C_β–C_β distances < 6Å is selected to form a model "active site". These 3 residues are included in the motif, and for each, there is a 50% chance of including one flanking residue. If no such triad is found in the monomer the task would fall back to Middle motif or Terminal motif with equal probability.
  - *For protein:small-molecule complexes*: A number of residues n is selected from $\mathcal{U}(1, 7)$. A candidate set of residues is then constructed from the set of the $n + 2$ closest residues to the small molecule union the set of any residues within 2Å of the n^{th} closest residue. n residues are then selected at uniform from this candidate set.

During training, the structure module blocks are not allowed to update the positions of the motif residues.

Losses: RFDiffusionAA was trained with a loss comprising two terms that closely follows the loss used in RFdiffusion:

$$\mathcal{L}_{\text{Diffusion}} = \mathcal{L}_{\text{Frame}} + w_{\text{2D}}\mathcal{L}_{\text{2D}},$$

$\mathcal{L}_{\text{2D}}$ is the same as that used in Section 2.10.5, with the modification that motif residues are not included in its calculation.

Where $\mathcal{L}_{\text{Frame}}$ is exponentially weighted over the intermediate structure module outputs, increasing towards the end of the network:

$$\mathcal{L}_{\text{Frame}} = \frac{1}{\sum_{i=0}^{I-1}\gamma^i}\sum_{i=1}^{I}\gamma^{I-i}d_{\text{Frame}}(x^{(0)}, \hat{x}^{(0),i})^2$$

where $\hat{x}^{(0),i}$ is the i^{th} structure block output and d_{Frame} is a weighted mean squared error which includes clamping on displacement.

$$d_{\text{Frame}}(x^{(0)}, \hat{x}^{(0)}) = \sqrt{\frac{1}{L}\sum_{l=1}^{L}\left(w_{\text{trans}}\min(\|z_l^{(0)} - \hat{z}_l^{(0)}\|_2, d_{\text{clamp}})^2 + w_{\text{rot}}\|I_3 - \hat{r}_l^{(0)\top}r_l^{(0)}\|_F^2\right)},$$

where w_{trans} and w_{rot} are weights on the rotation and translation distances, and d_{clamp} is a maximum distance above which translation distances are clamped. Note that the translation distance is only clamped p_{clamp} of the time.

The forward noising process uses the same functional form as RFdiffusion with the same parameter set. For further details on the noise schedule see [132].

Training time: RFDiffusionAA trained to convergence when initialized from RFAA weights in 10 epochs. This took 8 days on 8 NVIDIA A6000 GPUs.

Table 2.14: RFDiffusionAA training hyperparameters.

Parameter name	**Value**
Crop size	256
Pseudo-batch size	48
w_{trans}	0.5
w_{rot}	1.0
w_{2D}	1.0
d_{clamp}	10
p_{clamp}	0.9
Structure block iteration decay rate γ	0.99
Learning rate	0.0005, No warm-up. Decay learning rate by 0.95 after every 10000 optimization steps.
Examples per epoch	25600
Number of diffusion timesteps (T)	200
Variance schedule for translations	$\beta^{(t)} = \beta^z_{min} + (\frac{t}{T})(\beta^z_{max} - \beta^z_{min})$ with $\beta^z_{min} = 0.01$ and $\beta^z_{max} = 0.07$.
Variance schedule for rotations	$\sigma_t = \sigma_{min} + \frac{t}{T}\beta^r_{min} + \frac{1}{2}(\frac{t}{T})^2(\beta^r_{max} - \beta^r_{min})$, with $\sigma_{min} = 0.02$, $\beta^r_{min} = 1.06$, and $\beta^r_{max} = 1.77$
Probability of motif being contiguous or discontiguous	0.5
Probability of providing self-conditioning information	0.5
Coordinate scaling	0.25

2.10.9 In Silico Design Methods with RFdiffusionAA

In this subsection, we will describe the *in silico* methods we used to benchmark the ability of RF*diffusion* AA to generate small molecule binders and the methods used to design the proteins that were characterized experimentally.

Ligand Contact Potential

In RFdiffusion, it was shown that using a contact potential which biased trajectories towards forming interactions between protein chains improved success rates for generation of symmetric oligomers. We implemented a similar potential in RFdiffusionAA. These heuristic potentials update the C_α coordinates after each denoising step: $x^{(t-1)} = x^{(t-1)} + g(t)\nabla_{x^{(t)}} P(x^{(t)})$. Further motivating details can be found in [132]. The ligand contact potential we use is described below.

Denoting the coordinates of a ligand with K atoms by $s = \{s_k\}_{k=1}^{K}$ and the C coordinates of a protein by $z = [z_1, \ldots, z_L]$:

$$P_{\text{contact}}(z, s) = \sum_{1 \leq l \leq L} \text{Switch}(\min_{1 \leq k \leq K} \|z_l - s_k\|_2^2)$$

This potential is then multiplied by a time-dependent guide scale $g(t)$. In the *in silico* benchmark and bilin binder design cases a linear guide scale is used $g(t) = \frac{t}{T}$.

Small Molecule Binder Design Benchmark

We chose four small molecule targets to benchmark based on the following criteria: 1) the molecules are diverse in size and topology with respect to each other 2) two molecules are found in the training set and two molecules are novel to the training set (< 0.5 Tanimoto similarity to closest ligand in training; deposited after our training date cutoff). The chosen molecules are (by PDB 3-letter codes) FAD, SAM, IAI and OQO (conformations from PDB IDs: 7bkc, 7c7m, 5sdv, 7v11). For each target, we generated 400 distinct diffusion trajectories, only providing the small molecule conformation and no privileged information about the

backbone or any side chains in the native structure. For each generated backbone, we assign 8 sequences using LigandMPNN and predict all 8 sequences using AF2 in single sequence mode. We measure the RMSD of the AF2 predicted backbones compared to the design models to evaluate the probability of the backbone forming.

Rosetta Calculations on Diffused Structures

GALigandDock was run in "eval" mode (as done in subsection 2.10.7), to indicate that the ligand and any residues within a heavy atom contact distance of less than 8Å can move. We initialize the structure with the designed scaffold after the sequence has been assigned with LigandMPNN. LigandMPNN also predicts sidechain orientations which are used to initialize the sidechains for our energy calculations. The energy of the conformation is minimized using GALigandDock's generalized ligand potential and a harmonic constraint on the starting coordinates to prevent large movement. The pose is then scored using the generalized ligand potential in Rosetta.

Rosetta Ligand-Aware Relax

Rosetta metrics such as "ddG" and "contact molecular surface" to computationally score protein and ligand binding were applied for the designs generated with RFdiffusionAA and following sequence design. The generated protein and ligand complex models were relaxed using Rosetta prior to scoring with the Rosetta generic potential [94]. Harmonic potential on the protein Ca coordinates and the distances of selected pairs of protein backbone and ligand atoms were added as restraints to the generic scoring function. The estimated binding free energy "ddG" was calculated by taking the difference of *holo* and *apo* state Rosetta scores. The area of the target small-molecule packed by protein atoms was calculated using the Rosetta "contact molecular surface" metric [28]. This metric uses the contact distance to re-weight the contacting surfaces.

Assessing Diversity of Designs

Diversity was assessed by generating 100 designs for the four *in silico* benchmark ligands using RFdiffusionAA without use of a potential. In addition, 100 designs were generated with RFdiffusion for the unconditional case as well as for scaffolding a 20 residue motif from PDB ID 5TRV used as a benchmark case in [132]. All designs were 150 residues in length. For each of the six design cases, 100x100 pairwise TMAligns were performed. Agglomerative clustering with the distance metric $1 - TMscore$ was performed using the complete linkage criterion for various distance thresholds using $\text{scikit}-\text{learn}$ [96]. This ensures that each member of any given cluster has a TMScore to every other member of the cluster at least as high as the clustering threshold, which was swept from 0 to 1 in 400 evenly spaced increments. The results are shown in Fig S9: left shifted curves correspond to higher diversity.

Assessing Novelty of Designs

The 400 designs (without potential) for ligands FAD and SAM were TMAligned to the training dataset, and for each design, the highest scoring hit, the highest scoring hit with the same ligand, and the mean TMScore are recorded. The results are shown in Fig. S2.13. As expected, on average, designs are more similar to training data examples with the same ligand due to certain conserved binding modes (such as the Rossman fold for FAD). Yet, the distribution of Max TM-scores in Fig. S2.13B shows that the designs are not merely memorized examples from the training dataset for the target ligand, as the maximum TM scores to the training set are higher than TM scores to the training set containing the same ligand. The median maximum TM score of designs to a training example containing the same ligand is 0.61 (FAD) and 0.62 (SAM). The other two molecules in our benchmark were not present in the training set.

Bilin Binders

2776 designs across lengths 100, 150, and 200 were generated possessing the bilin-CARD motif at random locations at least 10 residues from either terminus. Half used a contact potential with linear decay. From those designs, 328 unique backbones were selected on the

basis of having AF2 RMSD < 2, AF2 motif RMSD < 1, AF2 PAE < 5, Rosetta ddG $<$ -7.5, and Contact Molecular Surface > 300 [28]. These 328 designs were then clustered into 100 groups to maximize the minimum intra-cluster TMScore and the design with the lowest AF2 RMSD was selected from each cluster.

Heme Binders

Heme-substrate model preparation The heme model used for the design of proteins with an open substrate pocket represents a transition state of the C-H abstraction reaction by the ferryl oxygen from the methoxy group of anisole (i.e., anisole *O*-demethylase activity). While the subsequent design methods are not necessarily expected to yield heme enzymes with that particular activity, the chosen model represents a chemically sound heme-substrate complex with a generic reactant structure shape and size. An additional *para*-phenyl group was added to anisole to increase the likelihood of generating protein backbones with surface-accessible substrate binding pockets.

The heme model was built by locating the *para*-phenylanisole methoxy C-H abstraction transition state based on an axially methanethiolate-ligated (as a mimic for cysteine) heme ferryl intermediate in quartet state.All calculations were performed using Gaussian 16 software.[49] Structural optimizations and frequency calculations were performed with B3LYP-D3 method along with 6-31G(d) basis set and the SDD ECP on Fe atom. D3 dispersion correction was applied using the Becke-Johnson damping function.[53] Solvent effects of water were included using the CPCM solvation model during optimization. Frequency calculations were performed to confirm whether the structure is a minimum or a transition state. Intrinsic reaction coordinate (IRC) analysis was used to confirm that the obtained transition states connect the correct minima.

Conformer library of the transition state was created based on sampling the dihedral angles within the two propionic acid groups, the rotation of substrate above the heme plane, and the two ether C-O bonds. Conformational diversity of the transition states was sampled using a frozen coordinate conformer sampling script (`https://github.com/ikalvet/`

`frozen-conf-xtb.git`) using the GFN2-XTB semiempirical QM method[14] for energy evaluation. 5000 conformers were saved.

The generated conformers were initially saved as XYZ files that were subsequently converted to MOLfiles using OpenBabel [93]. The bonding information in the MOLfile was manually inspected to ensure that the entire structure is represented as a single fragment, and edited, if necessary. Thereafter, mol2params.py script, available within Rosetta, was used to convert the MOLfile to a Rosetta-compatible .params file. The partial charges of the carboxylate oxygen atoms of the propionate groups were adjusted in the params files from -0.74 to -1.24 to increase the likelihood of H-bonds being created with these atoms during Rosetta design and relax.

Ligand model selection for diffusion A subset of 55 conformers from the thousands of generated ligand conformers were used as inputs for RFdiffusionAA. The selected heme-substrate complex models are intended to serve the purpose of guiding the diffusion trajectories towards creating a heme binding pocket with more of a top-open substrate access, akin to unspecific peroxygenases and cytochrome P450's. The conformers were selected based on three criteria:
1) Clustering the conformers based on structural similarity within 0.5 Å RMSD and selecting representative examples from each cluster (Fig. S2.25A).
2) Excluding conformers with the substrate pointed towards the carboxylate groups of heme. This was done to avoid creating an opening for a relatively hydrophobic substrate near possibly the most polar part of the heme binding site. Conformers with the C28-FE1-O5-C47 absolute dihedral angle greater than 50° were selected (Fig. S2.25B).
3) Excluding conformers where the substrate lies close to parallel against the heme plane. This was done to avoid creating heme binding sites with limited vertical space. Conformers with the FE1-O5-C47 angle greater than 140° were selected (Fig. S2.25C).

Heme binding site design To generate input structures for RFdiffusionAA, the selected conformers were aligned to the HEM ligand in the crystal structures of cytochrome P450

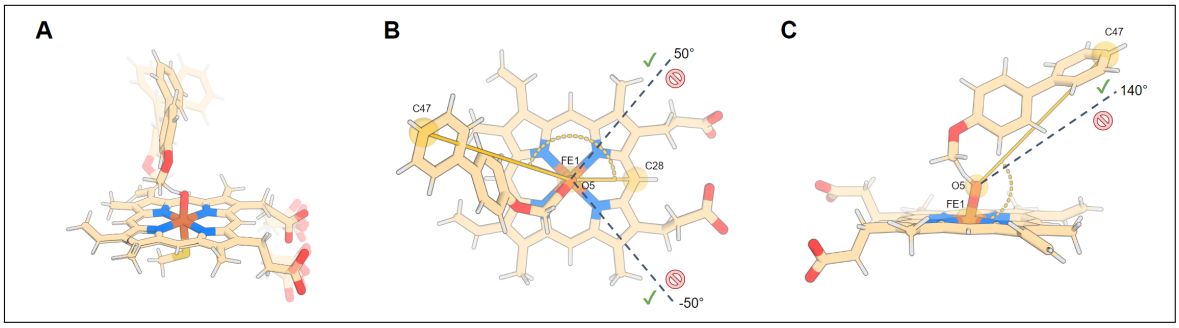


Figure 2.25: Selection of heme-substrate complex conformers as inputs for RFdiffusionAA. **A.** Representative conformers picked from clusters of similar conformers. **B.** Excluding rotamers where the substrate molecule is close to the carboxylate groups of heme. **C.** Excluding rotamers where the substrate molecule lies too close to parallel with the heme plane.

peroxygenase CYP152K6 (PDB: 6fyj), and unspecific peroxygenase *Hsp*UPO from *Hypoxylon sp. EC38* (PDB: 7o2g). Four orientations of heme were sampled based on the rotation around the S-Fe bond at 90° intervals, and PDB files were created representing each heme-substrate conformation in each of the sampled orientations. The heme-coordinating cysteine residues from these proteins were used as input motifs for RFdiffusionAA (Cys-15 for UPO and Cys-365 for P450). Furthermore, the selected motif residues were required to be outside of the first and the last 30 positions of any generated protein scaffolds. Protein backbones were generated with 150-210 amino acids in length. The resulting input 'contig' string based on the UPO motif is "30-150,A15-15,30-150".

An example configuration file ('config.yaml') for generating protein backbones around heme model with substrate, and UPO cysteine motif:

#####Start config.yaml#######

defaults:

- aa

diffuser:

T: 200

inference:

```
    num_designs: 30
    model_runner: NRBStyleSelfCond
    ligand: "HBA"

  model:
    freeze_track_motif: True

  contigmap:
    contigs: ["30-150,A15-15,30-150"]
    length: "150-210"

  potentials:
    guiding_potentials: ["type:ligand_ncontacts,weight:1"]
    guide_scale: 2
    guide_decay: cubic
#####End config.yaml#######
```

From each unique input conformer, 20-40 diffusion trajectories were spawned. A substrate contact potential was applied following either a quadratic or cubic decay to guide the trajectories towards creating proteins where the heme-binding site is sufficiently buried. With no ligand contact potential applied, a vast majority of diffused protein backbones lacked a well-buried heme binding site. On the other hand, using linear decay of the guiding potential led to significant clashes between the ligand and the scaffold. A total of ∼10000 backbones were generated through the combination of the different input motifs, conformers and contact potentials. From the generated protein backbones we filtered out globular proteins with desirable binding pocket qualities based on features describing ligand burial, substrate exposedness, lack of clashes between the ligand and protein backbone, loop content, distance of termini from the ligand, radius of gyration, and the length of the longest helix.

As part of the scaffold generation process, we also explored further diversification of the selected backbones using partial diffusion (fixed length global backbone remodeling)[123] or RoseTTAFold joint inpainting (variable length loop remodeling)[127]. Partial diffusion allows

subsampling the structural space roughly defined by the input scaffold. With RoseTTAFold joint inpainting (RF*joint*) we re-generated loop regions of the diffused backbones while allowing variable length loop substitutions. The loop regions were selected based on geometric distance from the ligand heavyatoms and the motif residue. Sequence design was enabled on all non-motif positions during the inpainting process. For each input scaffold, 5 diversified backbones were generated with both of these methods.

ProteinMPNN[37] was thereafter used to generate amino acid sequences optimal for each of the selected backbones. With the sequence design we reasoned that due to a significant degree of flexibility in the heme-substrate complex it would be beneficial to first identify sequences that are more likely to fold correctly before focusing on designing the protein-ligand interactions. Therefore, we then used ProteinMPNN to generate 15 sequences for each of the backbones while keeping the coordinating CYS fixed. Three sampling temperatures (0.1, 0.2, 0.3) were used, with 5 sequences generated at each, and introduction of additional cysteines was disallowed. Generated sequences were analyzed using single-sequence AlphaFold2 prediction (model 4 with 3 recycles) to identify the sequences that do fold into the desired shape. Successful sequences were selected based on AF2 metrics of plDDT > 87.0, Cα RMSD $<$ 1.5 Å, and CYS sidechain RMSD $<$ 1.2 Å. The selected backbones were then subjected to ligand binding site design by using either Rosetta FastDesign[7], or iterative application of ligandMPNN and Rosetta FastRelax[16]. The designable positions were selected based on distance from the heme-substrate complex heavyatoms, with residues with Cα atom within 8 Å from any ligand heavyatom considered for design. Geometric constraints were applied to the heme-CYS interaction using the Rosetta AddOrRemoveMatchCsts mover. Successful designs were selected based on metrics describing protein-ligand contacts (Rosetta ddG, contact molecular surface, SASA, H-bonds to acceptors). An additional round of ligandMPNN sequence design was performed at the positions flanking the binding pocket, with only those in the window of 8-12 Å from the ligand enabled for redesign. The structures of the resulting sequences were predicted using AlphaFold2 (with tighter cutoffs of Cα RMSD $<$ 1.2 Å and CYS sidechain RMSD $<$ 1.0 Å applied). Finally, the heme-substrate model was re-aligned into the AF2-predicted model structure, and relaxed with Rosetta FastRelax. The

relaxed models were re-evaluated based on the same protein-ligand interaction metrics as used before. Upon final manual inspection of the generated designs, 168 were selected for experimental testing.

The described design pipeline is available for download at: `https://github.com/ikalvet/heme_binder_diffusion`

Digoxigenin Binders

To demonstrate the capability of RFdiffusionAA in designing *de novo* binders for small molecules with the ligand information alone, we designed digoxigenin-binding proteins. We used a model conformation of the target ligand digoxigenin as an input to generate 25,000 diffused backbones using RFdiffusionAA. The configuration and parameters for generating the backbones are listed below.

RFdiffusionAA inference job was run using the following command: The diffusion outputs were subsequently filtered based on the radius of gyration, secondary structure content, solvent accessible surface area (SASA) of the small molecule, and the number of contacts between the protein backbone and ligand atoms. The selected backbones underwent ligand pose-aware sequence design by applying iterative cycles of LigandMPNN and Rosetta FastRelax ([7],[16], sequence design method described in [79]). Eight sequences were sampled per backbone with LigandMPNN, and AF2 was used to predict the protein structures. The selected designs with high AF2 pLDDT (>80) and low backbone-RMSD (<1.5Å) were used to re-dock the ligand digoxigenin using ChemNet. We reasoned that by employing ChemNet, a deep-learning fixed backbone ligand docking method, updated ligand positions would contribute in sampling more sequences from LigandMPNN[79]. We used the docking outputs with high ChemNet confidence (>80 ChemNet pLDDT; to filter out poor predictions from ChemNet) to perform a second round of sequence design with LigandMPNN. Finally, we conducted filtering of the designs based on AlphaFold2 and Rosetta metrics. The designs chosen for experimental testing exhibited high AF2 pLDDT (>80), low backbone-RMSD between design model and AF2 prediction (<2 Å), low Rosetta ddG (<-30), and formed at least one hydrogen bond

with the ligand. Given that our experimental screen involved digoxigenin covalently linked to biotin, we also excluded designs where the digoxigenin-linker atoms were deeply buried within the protein (Rosetta atomic_depth < 30). This resulted in 4,416 designs to experimentally screen.

2.10.10 Experimental Methods

Bilin Binders

Initial Screen of Bilin binding designs Synthetic genes encoding the RF Diffusion designs were ordered as eBlocks gene fragments in a 96-well plate format (Integrated DNA Technologies) and ligated into modified pET-29b as in [[37]]. The resulting plasmid arrays were transformed into *E. coli* BL21(DE3) harbouring pCOLADuet-*cpcEF-pebS*-HO1 ([8]). This plasmid encodes the biosynthetic machinery required to synthesize phycoerythrobilin (PEB) as well as a bilin-protein lyase, CpcEF, which is required to attach PEB to the CXRD bilin binding motif included in all the protein designs. PEB-binding positives in the 96-well plate were easily identified by their pink color, and were further characterized by scanning the plate using UV-Vis absorption (250-800 nm - Fluostar Omega) and fluorescence (excitation: 460, 520, and 630 nm; Emission filters: 530BP30, 605DF50, and 695DF55 - Amersham Imager 600).

Large scale expression and purification Positives from the initial screening were scaled up in 500 ml LB broth cultures in 2L baffled Erlenmeyer flasks at 37°C and 180 rpm in the presence of 50 μg/ml and 100 μg/ml, kanamycin and ampicillin, respectively. At an optical density at 600 nm (OD_{600}) of $\sim$ 0.6, IPTG was added to a final concentration of 1 mM to induce protein production and the cultures were switched to 18°C for 16 h. *E. coli* cell pellets were resuspended in 50 mM HEPES, 500 mM NaCl, 20 mM Imidazole (pH 7.6), disrupted by sonication, and the His-tagged biliproteins were purified from the cell extracts via immobilized metal affinity chromatography (IMAC) and subsequent size-exclusion chromatography (SEC).

Fluorescence analysis of bilin binding proteins Samples were diluted to OD $\sim$ 0.1 for recording fluorescence emission spectra (Horiba Fluorolog-3), using a 10 nm FWHM excitation source. An absorptance (1-T) spectrum was calculated for each sample (Agilent Cary 60), and the ratio of absorptance to fluorescence (areas under the graph) were used to estimate the relative fluorescence yield for each biliprotein, with CpcA-PEB arbitrarily set to 100%.

Large scale expression of PEB synthesis genes in presence and absence of bilin binding proteins The plasmid for expression of one of the de novo biliproteins from the main manuscript – C11 – and a plasmid capable of expressing large quantities of maltose binding protein (MBP) were transformed into *E. coli* (BL21(DE3)) harbouring a plasmid for synthesising PEB and expressing the proteins CpcE/F (bilin lyase). These were grown up in 500 ml of LB and induced between OD 0.6-0.8 using 1mM IPTG, expression was carried out for 16-18hrs at 16C. Cells were resuspended in lysis buffer (50 mM HEPES + 500 mM NaCl at pH 7.6 with protease inhibitor, DNase I and lysozyme) and sonicated to release the cellular content. After clarification – removal of cellular debris – by centrifugation at 76k xg, spectra of the supernatant were taken (Fig. S2.21A, B). Finally, the supernatant of the PEB only and PEB + C11 was purified via an IMAC column for C11 + PEB and the supernatant of PEB + MBP was purified via an amylose column. Spectra were taken of each of the eluents (Fig. S2.21B, D).

Heme Binders

Construction of pET29b(+) plasmids encoding heme binding protein variants Double-stranded DNA fragments encoding the designs and any variants thereof (codon-optimized for bacterial expression) were purchased from Integrated DNA Technologies (IDT) as eBlocks™ Gene Fragments. Following the Golden Gate cloning protocol using T4 DNA ligase and BsaI-HFv2 restriction enzyme (master mix #E1601, NEB),[133] the DNA fragments encoding design sequences and including overhangs suitable for a BsaI restriction digest were cloned into a custom pET29b(+) target vector containing lethal ccdb gene, and C-terminal SNAC[35] and hexahistidine tags (#191551, Addgene). This yielded final expressed sequences as: MSG <design> GSGSHHWGSTHHHHHH.

Small-scale screen for expression Assembled plasmids containing the designs were transformed into chemically competent *E. coli* BL21(DE3) cells by heat shock. DNA was incubated on ice with competent cells for 30 minutes, followed by 30 second heat shock at 42 °C, and 2 minute incubation on ice. 100 μL rich medium (super optimal broth with catabolite repression, SOC) was added to transformed cells and samples were incubated at 37 °C, 1050

r.p.m. on a shaking platform for 1 hour. The cells were subsequently transferred to 900 μL of LB medium containing 50 μg/mL kanamycin, and incubated on a shaking platform (1050 r.p.m) at 37 °C for 16 hours. Thereafter, 100 μL of the starter culture was transferred to 900 μL of TB-II medium containing 50 μg/mL kanamycin and the cultures were grown at 37 °C for 4 hours. Protein expression was induced by the addition of 2 mM IPTG and the cultures were incubated at 37 °C for 2 hours. Meanwhile, glycerol stocks were prepared by mixing 100 μL of the starter culture in LB media with 100 μL of 50% glycerol and stored at -80 °C. The cell pellets were harvested by centrifugation at 4,000 g for 10 minutes and lysed with BugBuster® lysis reagent containing 0.01 mg/mL deoxyribonuclease and 0.1 mg/mL lysozyme, using a ultrasonication in a microplate horn (40% amplitude with 10 s on, 10 s off for a total of 4 min on-time). Lysate was collected by centrifugation at 4,000 xg for 20 minutes and analyzed by SDS-PAGE to identify solubly expressed protein variants. The variants not showing a band of over-expressed protein on the SDS-PAGE gel may have done so for a number of reasons: very low level of expression; insoluble protein; errors in the cloned DNA fragment. The precise modes of failure were not investigated at this stage of screening.

Small-scale heme-binding screen To clarified cell lysates containing overexpressed protein was added 9 μL of hemin solution (250 μM in 0.5 M aq. NaOH) to reach a final hemin concentration of 10 μM. The lysates were then applied to Ni-NTA resin (50 μL) that was equilibrated with wash buffer (50 mM KPi, 200 mM NaCl, 25 mM imidazole, pH 7.4). The resin was washed with 25 column volumes (CV) of wash buffer. Protein was eluted with 200 μL of elution buffer (50 mM KPi, 200 mM NaCl, 300 mM imidazole, pH 8.0). To remove non-specifically bound heme and most of the imidazole, 130 μL of the eluted protein solutions were thereafter loaded onto a 96-well PD MultiTrap G-25 desalting plate (Cytiva) equilibrated with a buffer containing 50 mM KPi and 200 mM NaCl at pH 7.2, and eluted by centrifugation at 800 xg for 2 minutes. As a control, the IMAC elution buffer alone was also eluted through the desalting column, and the resulting solution used as a background in the subsequent UV-Vis spectroscopic analysis. UV-Vis absorbance spectra of the desalted heme-loaded protein solutions were collected in half-area UV-STAR microplates (Greiner) using a platereader (BioTek Synergy Neo2) in the 250-700 nm range. Heme-binding was

qualitatively assessed based on the wavelength and the intensity of the Soret maximum of heme. A Soret maximum at 420-425 nm is indicative of CYS-ligated heme-binding in a hexacoordinate low-spin state (with the 6th coordination site most likely filled by endogenous imidazole), whereas a Soret maximum at 370-390 nm, along with a charge transfer band at 640 nm is indicative of a CYS-ligated pentacoordinate high-spin heme state.[114] The collected spectra are presented in Fig. S2.26-2.28.

Larger-scale expression and purification of heme-binding proteins 40 designs were selected based on the results of the small-scale heme-binding assay for expression scale-up and further characterization. Clonal variants of the designs were obtained by spreading stabs from the polyclonal glycerol stocks on LB-agar plates containing 100 μg/mL kanamycin and incubating the plates at 37 °C for 16 hours. Single colonies were picked, and the DNA fragments encoding the designs were amplified following a colonyPCR protocol using GoTaq® Green DNA polymerase master mix (#M7122; Promega) and T7 reverse and forward primers. The PCR products identified to contain DNA of appropriate size based on agarose gel (1.2%) electrophoresis with SybrSafe dye were sent to Sanger sequencing (GeneWiz/Azenta) for sequence-verification. Single colonies containing the correct design sequences were grown up in 5 mL LB media containing 50 μg/mL kanamycin, over 16 hours at 37 °C. 2 mL of the starter culture was used to inoculate 40 mL TB-II media containing 50 μg/mL kanamycin and the rest used for plasmid extraction following the Qiagen QIAprep MiniPrep protocol. The 40 mL cultures were grown at 37 °C for 4 hours, after which protein expression was induced with the addition of 1 mM IPTG, and the cultures were incubated at 37 °C for 2 hours. Pellets were harvested by centrifugation at 4,198 g for 8 minutes and resuspended in a lysis buffer containing 50 mM KPi, 200 mM NaCl, 25 mM imidazole, 0.01 mg/mL DNAse, 0.1 mg/mL lysozyme, and a Pierce protease inhibitor tablet. 200 μM hemin (from 12 mM stock in 0.5 M aq. NaOH) was added to the resuspended cells, cooled on ice. Lysis was immediately performed by ultrasonication (13 mm probe, 2.5 mins, 10s on, 10s off, 65% amplitude). Lysate was clarified by centrifugation at 15,000 xg for 20 minutes, and applied to Ni-NTA resin that was equilibrated with wash buffer (50 mM KPi, 200 mM NaCl, 25 mM imidazole, pH 7.4). The resin was washed extensively with 50 column volumes (CV) of wash

buffer. Protein was eluted with 1.2 CV of elution buffer (50 mM KPi, 200 mM NaCl, 300 mM imidazole, pH 8.0) and further purified via size exclusion chromatography (SEC) using a Superdex Increase 75 10/300 GL column (GE Healthcare) on ÄKTAxpress (GE Healthcare) instrument at 0.8 mL min-1 flow rate using a running buffer containing 50 mM KPi, 200 mM NaCl pH 7.2. The monomeric or smallest oligomeric fractions of each run were collected. The obtained chromatograms are presented in Fig. S2.29-2.30. UV/Vis spectra were collected of the heme-loaded protein samples after purification to determine the iron coordination number and to qualitatively compare heme loading levels in isolated proteins (as judged by the A_{Soret}/A_{280} ratio). The collected UV/Vis spectra of purified proteins are presented in Fig. S2.31.

The Cys/Ala knockout mutants of selected designs were produced by following the aforementioned Golden Gate assembly protocol, and transformed into *E. coli* BL21(DE3) cells as described above. After incubation in SOC media, the cells were spread on LB-agar plates containing 100 μg/mL kanamycin, incubated at 37 °C for 16 hours. Single colonies were picked and sequence-verified, and the larger-scale expression protocol was followed in most part. These proteins were purified without the addition of hemin prior to the lysis step, and were isolated as *apo* proteins.

Mass spectrometry analysis MS data for the designed proteins were acquired on an Agilent 1200series LC G6230B TOF LC-MS with an AdvanceBio RP-Desalting column (A: H_2O with 0.1% Formic Acid, B: Acetonitrile with 0.1% Formic Acid). The final protein concentrations were adjusted to 1-2 mg/mL in 50 mM KPi, 200 mM NaCl, pH 7.2. Subsequent data deconvolution was performed in Bioconfirm using a total entropy algorithm. All data are presented in Table S2.15.

Variable temperature spectrophotometric measurements To observe changes in the spectral properties of bound heme at increasing temperatures, UV/Vis spectra were measured of in vitro loaded holo-proteins using the Jasco Spec V750 spectrophotometer and a 10 mm pathlength cuvette. Spectra in the 230-700 nm range were collected at every 10

°C intervals between 25 °C and 95 °C. Temperature was increased at the rate of 5 °C min^{-1}, and spectra were acquired after the temperature had stabilized to within 0.5 °C of target temperature for 5 seconds. Measurements were performed with 20 μM solutions of purified holoprotein in KPi buffer (50 mM KPi, 200 mM NaCl, pH 7.2). The obtained results are presented in Fig. S2.14-2.17.

Circular dichroism spectroscopy To determine secondary structure and thermostability of the designs, far-ultraviolet circular dichroism (CD) measurements were carried out on a JASCO J-1500 instrument using a 1 mm pathlength cuvette. Samples of purified protein were prepared at 0.3-1.0 mg/mL in 50 mM KPi, 20 mM NaCl, pH 7.2. The temperature of the sample was scanned from 25 °C to 95 °C with full spectrum scans from 190 nm to 260 nm performed after each 10 degree increment. Protein concentrations were determined by absorbance at 280 nm, measured using a NanoDrop spectrophotometer (Thermo Scientific) using predicted extinction coefficients.[51] The obtained results are presented in Fig. S2.14-2.17.

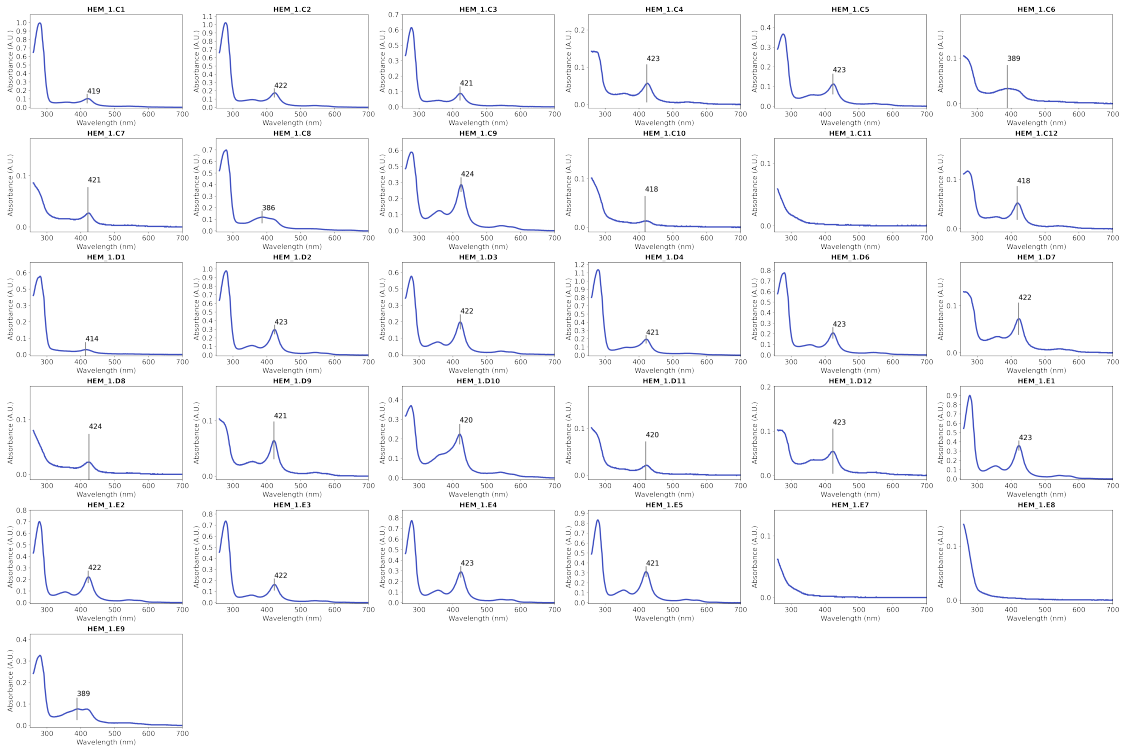


Figure 2.26: UV/Vis spectra collected from small-scale screening for heme binding, HEM_1.C1 to HEM_1.E9. Samples were prepared by adding 10 μM hemin to clarified cell lysates (grown from 1 mL cell cultures), purifying the proteins with Ni-NTA affinity chromatography, and eluting through PD MultiTrap G-25 desalting column.

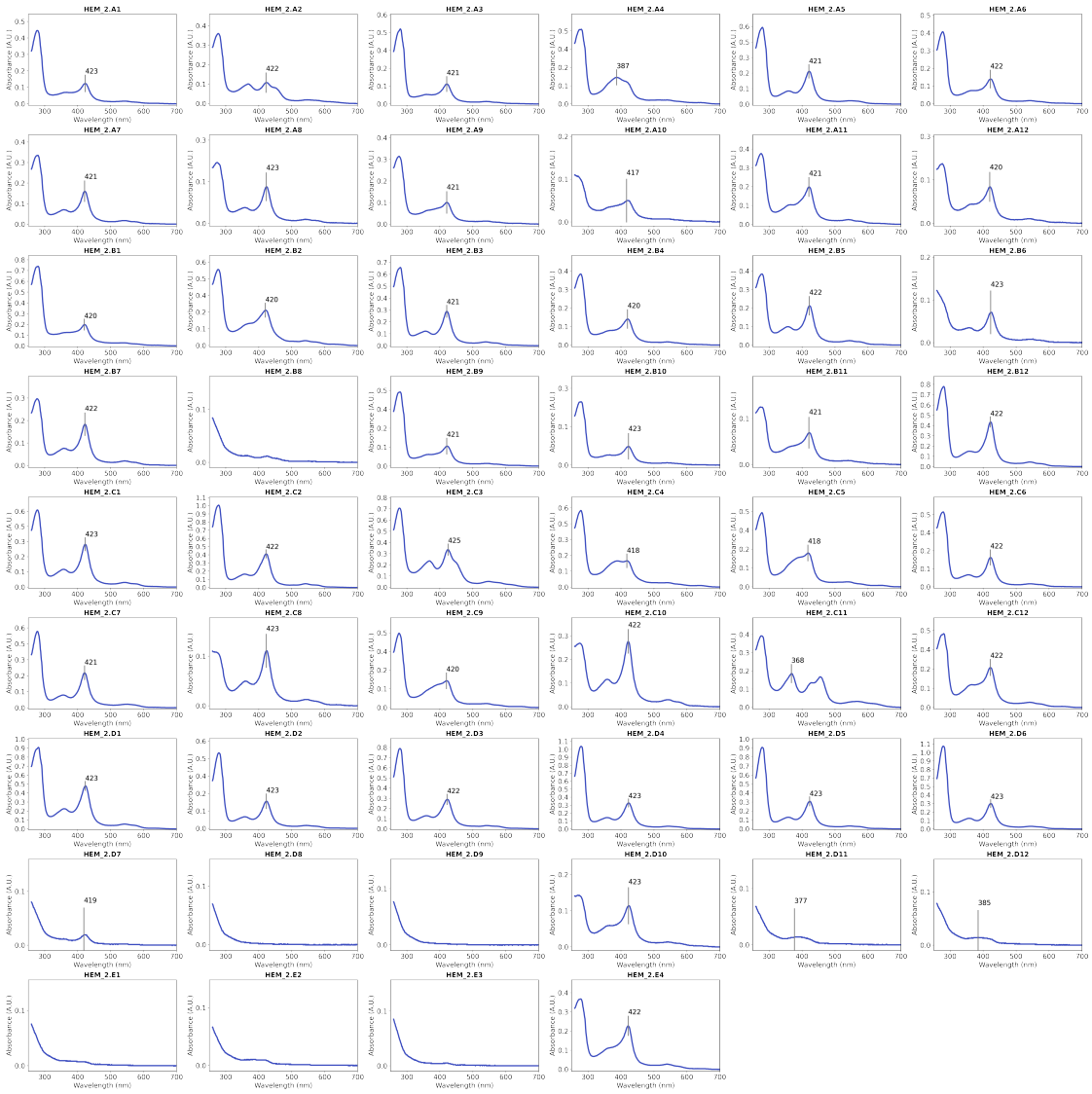


Figure 2.27: UV/Vis spectra collected from small-scale screening for heme binding, HEM_2.A1 to HEM_2.E4. Samples were prepared by adding 10 μM hemin to clarified cell lysates (grown from 1 mL cell cultures), purifying the proteins with Ni-NTA affinity chromatography, and eluting through PD MultiTrap G-25 desalting column.

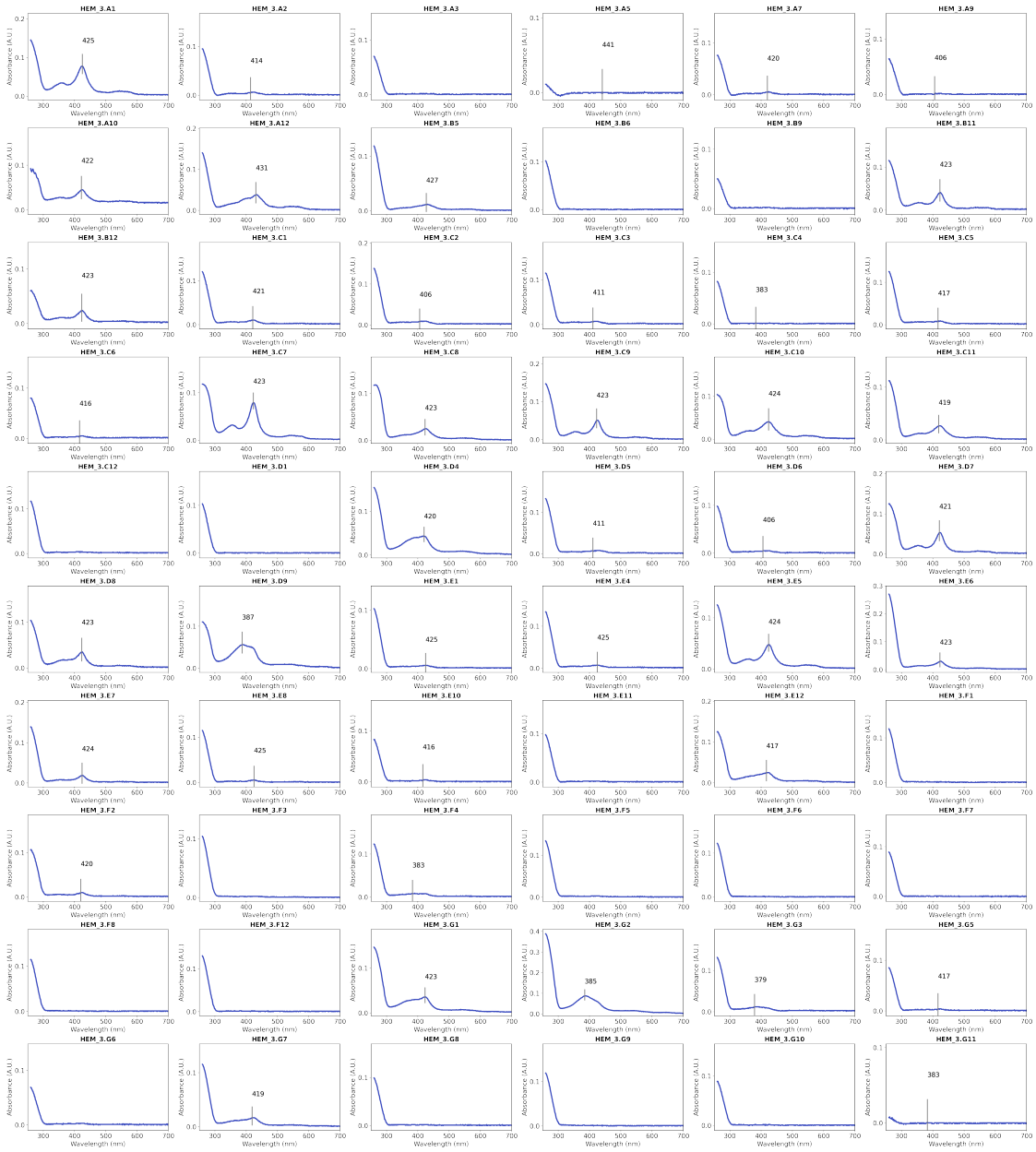


Figure 2.28: UV/Vis spectra collected from small-scale screening for heme binding HEM_3.A1 to HEM_3.G11. Samples were prepared by adding 10 μM hemin to clarified cell lysates (grown from 1 mL cell cultures), purifying the proteins with Ni-NTA affinity chromatography, and eluting through PD MultiTrap G-25 desalting column.

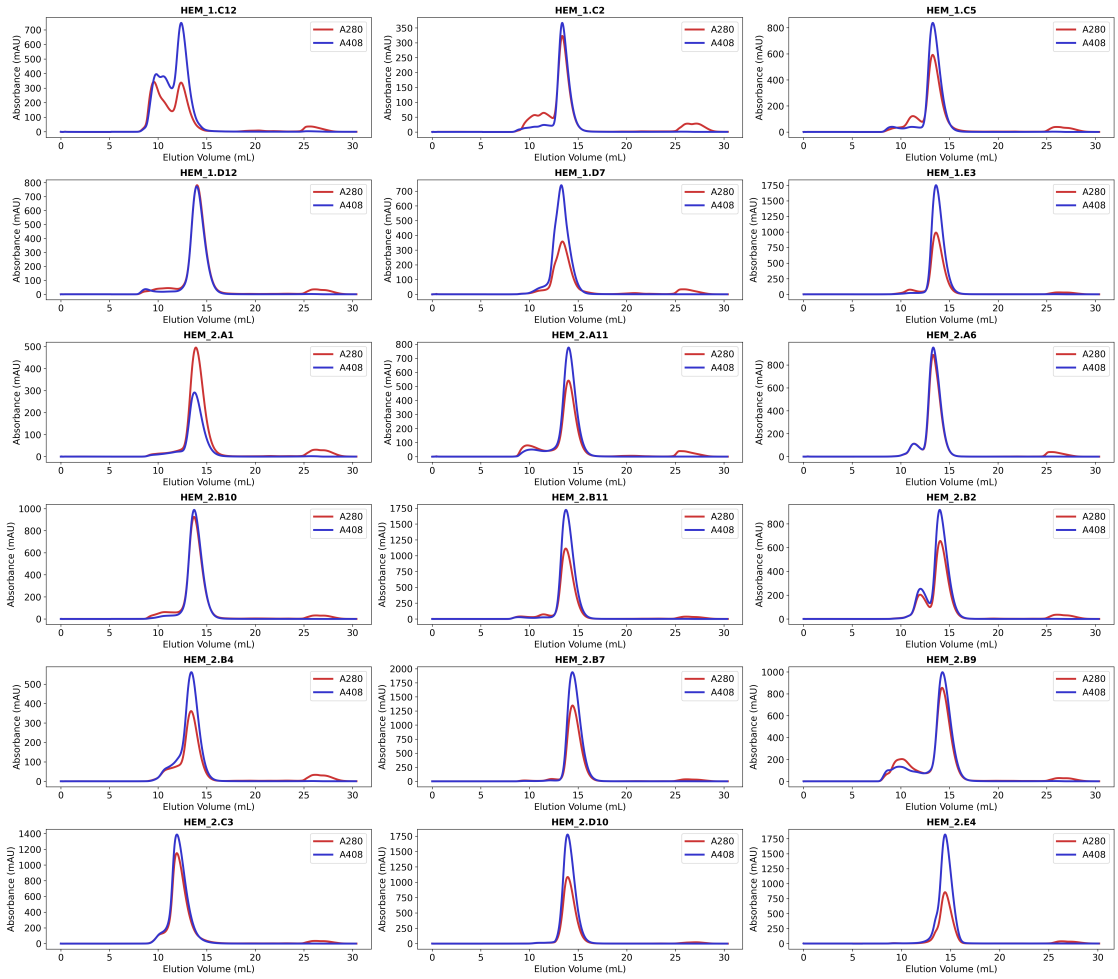


Figure 2.29: Size-exclusion chromatograms of heme-loaded proteins, HEM_1.C2 to HEM_2.D10. Data were collected using a Superdex Increase 75 10/300 GL column (GE Healthcare) in a buffer containing 50 mM KPi and 200 mM NaCl at pH 7.2. Void volume of the column is 8.5 mL. Blue chromatograms were obtained by following the absorbance at 408 nm, indicating elution of heme-containing species. Red chromatograms were obtained from absorbance at 280 nm.

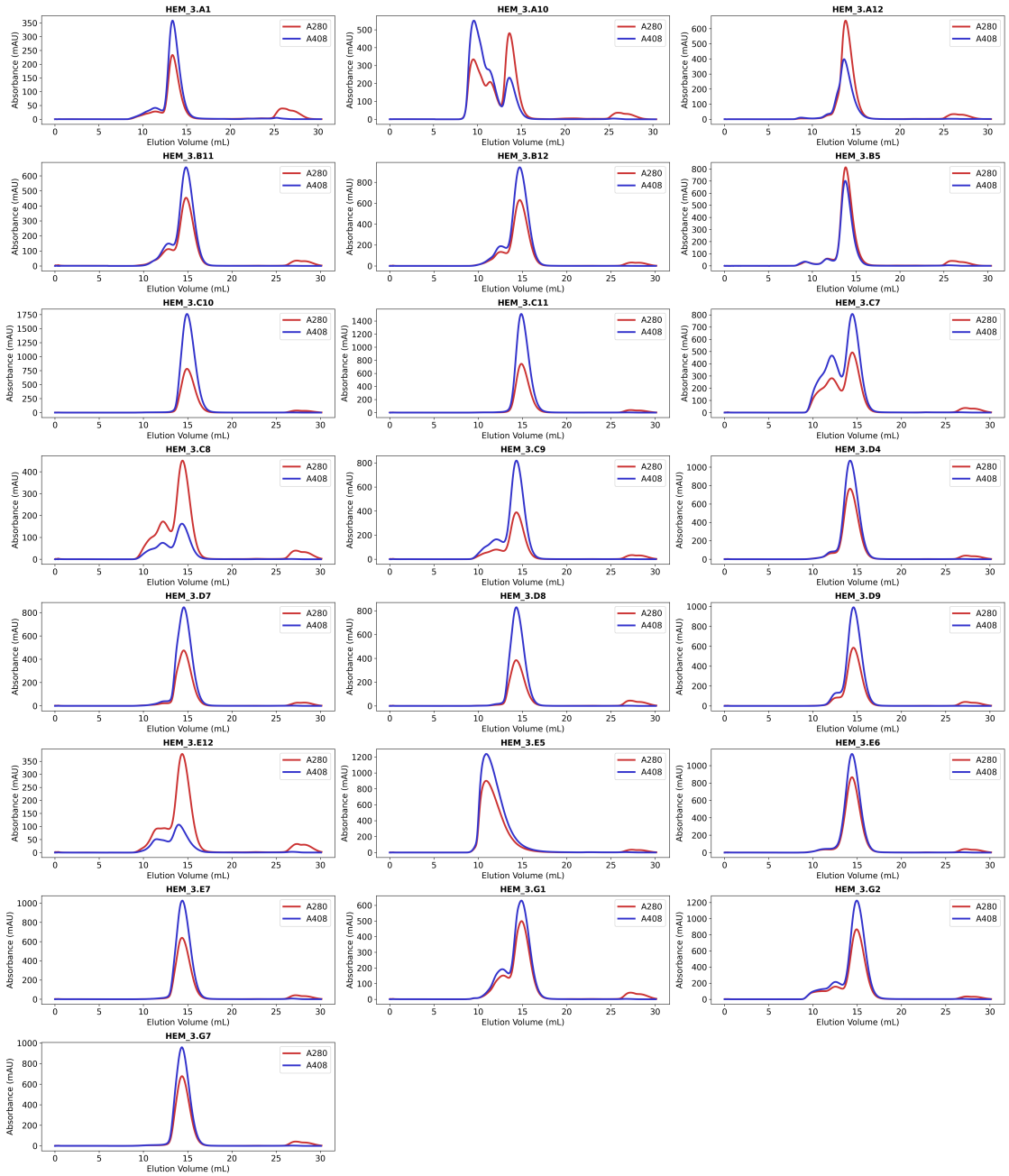


Figure 2.30: Size-exclusion chromatograms of heme-loaded proteins, HEM_3.A1 to HEM_3.G7. Data were collected using a Superdex Increase 75 10/300 GL column (GE Healthcare) in a buffer containing 50 mM KPi and 200 mM NaCl at pH 7.2. Void volume of the column is 8.5 mL. Blue chromatograms were obtained by following the absorbance at 408 nm, indicating elution of heme-containing species. Red chromatograms were obtained from absorbance at 280 nm.

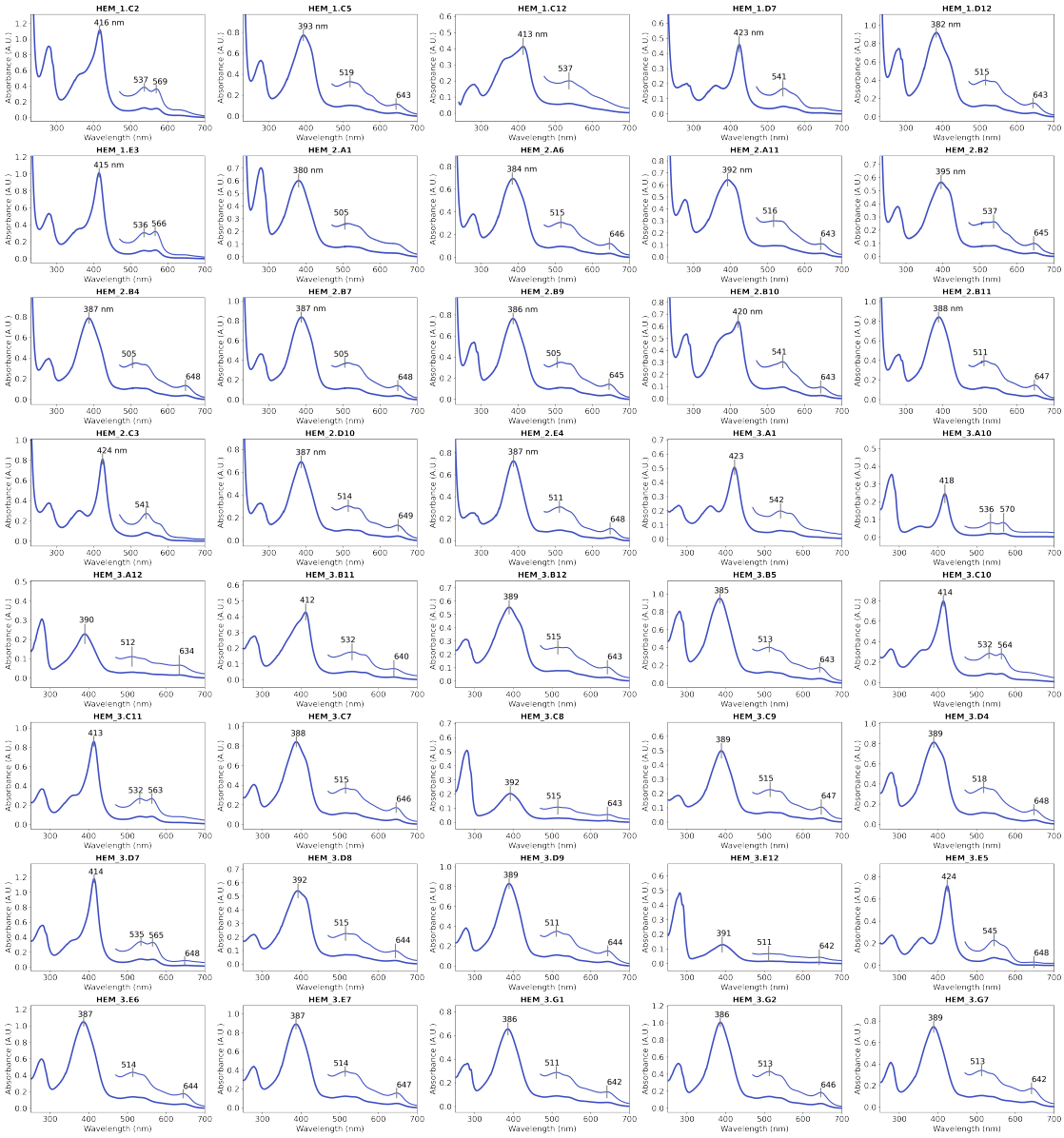


Figure 2.31: UV/Vis spectra of heme-loaded proteins. Inset shows the visible region at 3x magnification. Spectroscopic data of most designs is in agreement with CYS-ligated heme binding (either Soret maximum at ∼420 nm and Q band features at 540/570 nm for hexacoordinate low spin state, or Soret maximum at 370-390 nm and Q band features at 510/540 and charge transfer band at 640 nm for pentacoordinate high spin state). Spectra were recorded in a buffer containing 50 mM KPi and 200 mM NaCl at pH 7.2.

Table 2.15: Mass spectrometry data for diffused heme-binding proteins. In all cases the observed mass corresponded to the loss of N-terminal methionine (-130 Da).

Variant	**Expected Mass**	**Observed Mass**		**Variant**	**Expected Mass**	**Observed Mass**
HEM_1.C2	27185	27054		HEM_3.A12	21607	21476
HEM_1.C5	26735	26604		HEM_3.B5	24682	24551
HEM_1.C12	20524	20393		HEM_3.B11	21344	21213
HEM_1.D7	22406	22275		HEM_3.B12	21761	21630
HEM_1.D12	21959	21828		HEM_3.C7	23187	23056
HEM_1.E3	22940	22809		HEM_3.C8	23218	23087
HEM_2.A1	23983	23852		HEM_3.C9	23179	23048
HEM_2.A6	23630	23499		HEM_3.C10	23580	23449
HEM_2.A11	22449	22319		HEM_3.C11	23542	23411
HEM_2.B2	22750	22619		HEM_3.D4	21383	21252
HEM_2.B4	22449	22318		HEM_3.D7	24155	24024
HEM_2.B7	17507	17377		HEM_3.D8	21197	21066
HEM_2.B9	16938	16807		HEM_3.D9	21326	21195
HEM_2.B10	17123	16992		HEM_3.E5	21084	20953
HEM_2.B11	22889	22758		HEM_3.E6	23364	23233
HEM_2.C3	19173	19042		HEM_3.E7	23226	23095
HEM_2.D10	22007	21876		HEM_3.E12	23057	22926
HEM_2.E4	16922	16791		HEM_3.G1	22702	22571
HEM_3.A1	24975	24844		HEM_3.G2	23108	22977
HEM_3.A10	24457	24326		HEM_3.G7	22482	22351

Crystallographic data Protein sample of **HEM_3.C9** for crystallography was prepared following the procedure outlined in paragraph "Larger-scale expression and purification of heme-binding proteins". The holoprotein was purified using Ni-affinity and size exclusion chromatography. The C-terminal hexahistidine tag was left intact. The *holo* **HEM_3.C9** was crystallized at 20 mg mL^{-1} in purification buffer (50 mM KPi, 200 mM NaCl, pH 7.2). The concentration of the protein sample was determined based on its absorbance at 280 nm, and the molar extinction coefficient (11460) and molecular weight (23179) used for calculating the concentration were obtained with ProtParam Expasy.[51]

All crystallization experiments were conducted using the sitting drop vapor diffusion method. Crystallization trials were set up in 200 nL drops using the 96-well plate format at 20 °C. Crystallization plates were set up using a Mosquito LCP from SPT Labtech, then imaged using UVEX microscopes and UVEX PS-256 from JAN Scientific. Diffraction quality crystals formed in 0.1 M Citric acid pH 5.0, and 3.2 M Ammonium sulfate (JCSG+ crystallography screen, well F2). Crystals were flash-frozen in liquid Nitrogen before sending out to analysis at the synchrotron.

Diffraction data was collected on AMX at the National Synchrotron Light Source II. X-ray intensities and data reduction were evaluated and integrated using XDS[73] and merged/scaled using Pointless/Aimless in the CCP4 program suite[135]. Structure determination and refinement starting phases were obtained by molecular replacement using Phaser[91] using the designed model for the structures. Following molecular replacement, the models were improved using phenix.autobuild[3] using simulated annealing. Structures were refined in Phenix[3]. Model building was performed using COOT[44]. The final model was evaluated using MolProbity[134]. Data collection and refinement statistics are recorded in Table S2.16. Data deposition, atomic coordinates, and structure factors reported in this paper have been deposited in the Protein Data Bank (PDB), http://www.rcsb.org/ with accession code 8vc8.

Table 2.16: Data collection and refinement statistics for co-crystal structure of HEM_3.C9.

	HEM_3.C9
PDB accession number	8VC8
Wavelength	0.92009
Resolution range (Å)	28.06 - 1.8 (1.85 - 1.8)
Space group	P 2_1 2_1 2_1
Unit cell dimensions a, b, c, (Å) α, β, γ (°)	 37.987, 39.622, 119.239 90, 90, 90
Unique reflections	16611 (1242)
Multiplicity	4.3 (4.1)
Completeness (%)	95.57 (88.59)
Mean I/sigma(I)	8.84 (1.01)
Wilson B-factor (Å^2)	27.26
R-merge	0.094 (1.420)
R-pim	0.050 (0.792)
$CC_{1/2}$	0.999 (0.746)
Reflections used in refinement	16611 (1242)
Reflections used for R-free	1650 (120)
R-work	0.1893 (0.3221)
R-free	0.2319 (0.3411)
Number of non-hydrogen atoms	1610
macromolecules	1500
ligands	48
solvent	62
Protein residues	197
RMS(bonds)	0.015
RMS(angles)	1.4
Ramachandran favored (%)	98.97
Ramachandran allowed (%)	1.03
Ramachandran outliers (%)	0
Average B-factor (Å^2)	33.39
macromolecules	33.19
ligands	32.27
solvent	39.18

Digoxigenin Binders

The selected designs were ordered as synthetic oligonucleotides (Twist Bioscience) and transformed into *S. cerevisiae* EBY100 strain as a pooled library, using a previously described protocol [28]. EBY100 culture was grown in C-Trp-Ura medium supplemented with 2% (w/v) glucose (CTUG). For induction of expression, yeast cells initially grown in CTUG were transferred to SGCAA medium supplemented with 0.2% (w/v) glucose and induced at 30 °C for 16–24 h. Cells were washed with PBSF (PBS with 0.1% (w/v) BSA) and incubated with a solution containing 2 μM of biotinylated digoxigenin, streptavidin conjugated to PE (SA-PE, Invitrogen), and anti-myc antibody conjugated to FITC (Immunology Consultants Laboratory) for 40 minutes at room temperature. After incubation time, cells were washed with PBSF and resuspended before cell sorting. We performed fluorescent activated cell sorting (FACS, SONY SH800S) to collect cells with PE-signal which represents binding to biotinylated digoxigenin. We performed a second round of cell sorting with three different conditions of incubation, which were prepared by having different biotinylated digoxigenin concentrations. We identified three designs that showed enrichment for binding to biotinylated digoxigenin and SA-PE. We used flow cytometry (Attune NxT, Thermo Fisher) to analyze the binding signals (from biotin-digoxigenin and SA-PE) of each clone (Fig. S2.32).

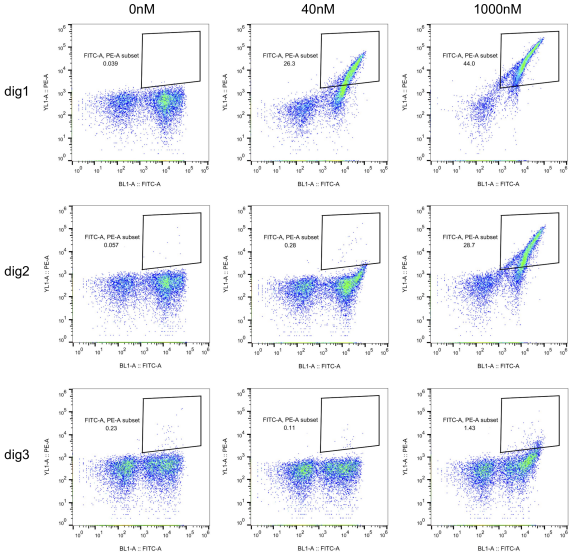


Figure 2.32: Binding signals of three digoxigenin binder hits analyzed by yeast display and flow cytometry. Binding signals of the samples with biotin-digoxigenin titrated (concentrations 0 nM, 40 nM, 1000 nM incubated with SA-PE) are shown in columns.

To characterize the binding affinity of the designs *in vitro*, we used Golden Gate assembly reaction with BsaI-HFv2 restrction enzyme (NEB) to clone the gene fragments of the potential hit sequences into a custom pET29b(+) target vector including a BsaI restriction site, lethal ccdb gene, and N terminal histidine tag for protein expression in *E. coli* [133]. This resulted in the final expressed sequence being MSHHHHHHSG-design-GS. We transformed the cloned plasmid into BL21(DE3) competent cells (NEB), and the *E. coli* cells were incubated for growth and expression in autoinduction medium for 16 hrs at 37 °C. The expressed proteins were purified using His-tag and Nickel affinity with IMAC, and further purification was achieved by using size-exclusion chromatography (SEC) in phoshpate buffered saline with 137 mM NaCl, 2.7 mM KCl and 11.9 mM phosphates (PBS, Fisher). Superdex Increase 75 10/300 GL column (GE Healthcare) was used with ÄKTAxpress (GE Healthcare) for SEC and we collected monodisperse fractions to further determine the binding affinity to digoxigenin. Equilibrium fluorescence polarization (FP) experiment was performed with decreasing protein concentrations when the concentration of AlexaFluor488-labeled digoxigenin (AF488-DIG) was fixed as 5nM in phosphate buffered saline (PBS) as done previously [122] (Fig. S2.33). Ligand binding of dig3 was too weak and could not be characterized *in vitro* (data not shown). We further performed equilibrium competition fluorescence polarization binding assays to dig1 and dig2 (Fig. S2.33). The concentrations of AF488-DIG and the binder were fixed (0.5 nM, 40 nM and 1 nM, 200 nM for dig1 and dig2, respectively), and fluorescence polarization values at increasing concentrations of label-free digoxigenin (Sigma-Aldrich) were measured.

Isothermal titration calorimetry (ITC) was carried out for the best binding design dig1 by injecting 256 μM label-free digoxigenin to 22.5 μM dig1 protein in 20 mM Tris, 100 mM NaCl, pH 8.0 buffer with 1.0% DMSO using the 19 injections with syringe rinsing protocol of MicroCal PEAQ-ITC (Malvern Panalytical). ITC binding isotherm curve is shown in Fig. S2.34. K_{d} was measured to be 343 +/- 109 nM and the binding stoichiometry was 1.02. From the SEC-MALS (Agilent 1200 HPLC, GE Superdex 75 -10x300 column, 20 mM Tris 100 mM NaCl, pH 8.0 buffer) experiment results we observed that the purified dig1 protein is majorly monomeric (Fig. S2.35). The concentration of dig1 for performing SEC-MALS experiment was 112.9 μM.

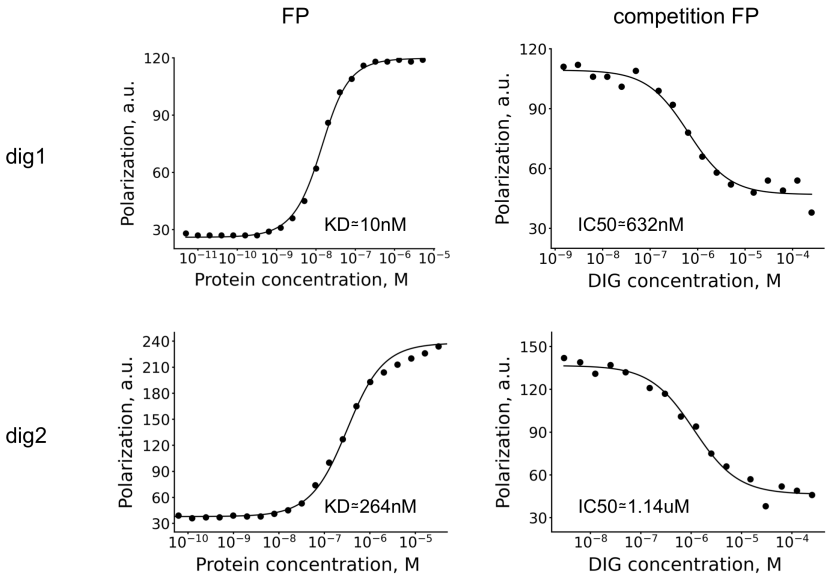


Figure 2.33: Equilibrium fluorescence polarization with AF488-DIG (FP) and competition fluorescence polarization with label-free DIG (competition FP). K_{d} for AF488-DIG and IC_{50} for DIG are estimated for the binders dig1 and dig2.

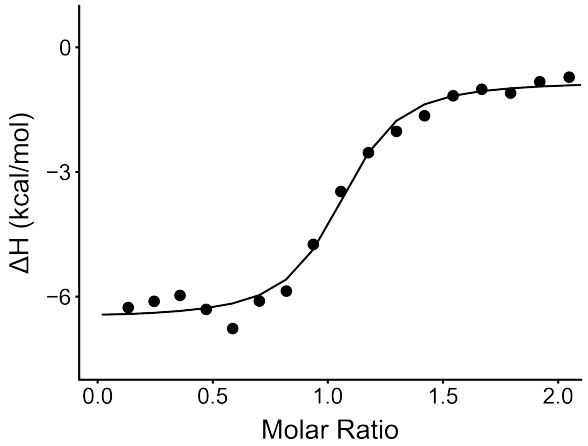


Figure 2.34: Secondary binding affinity dataset using ITC for the tighest binder dig1. K_{d} estimate: 343 nM +/- 109 nM. Binding stoichiometry: 1.02.

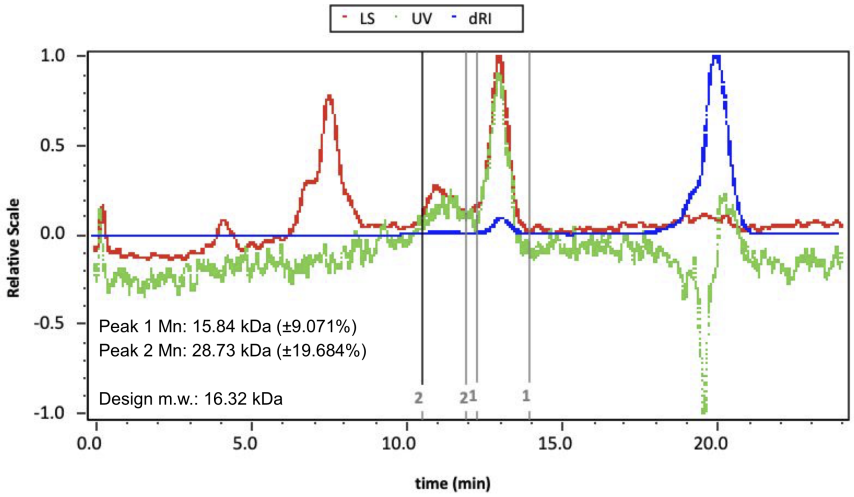


Figure 2.35: Determined molecular mass of dig1 from SEC-MALS experiment. The major peak1 had number weighted molecular mass (Mn) of 15.8 kDa which is close to the monomeric molecular weight of the design dig1 (16.3 kDa). The minor peak2 had a number weighted molecular mass of 28.7 kDa close to the dimeric molecular weight of dig1.

2.10.11 Figures and Statistics

Figures were generated using matplotlib[69] and seaborn[130] and appropriate statistical tests were performed with Scipy[126] when noted in figure legends. Outliers were removed from boxplots for clarity. Renderings of protein structures were created with PyMOL.

2.10.12 Supplementary Results

Ligand-Aware Protein Structure Prediction

In order to investigate whether or not the additional ligand context helps the RFAA model make better predictions of protein structure (Fig. S2.5), we filtered the dataset from subsection 2.10.7 down to only those items that RFAA predicts confidently (inter-chain PAE < 10.0). We hypothesize that in the cases that RFAA gets the ligand dock correct, it may predict more accurate structures of the protein binding pocket. In those cases in which RFAA fails to dock the ligand correctly, it is unlikely that the added ligand context will aid in protein structure prediction.

Comparing against RF2 We predict the same protein chains (those that RFAA predicts with high confidence) with RF2, which has no way to represent the bound ligand. This a safe comparison because the validation clusters between RFAA and RF2 are identical so the only large differences are the architectural differences described above and the ability to explicitly model ligands. We expect that comparisons to other structure prediction networks will be harder because of training dataset bias. We further filter the evaluation set by confident prediction from RF2 (protein pLDDT > 80), and end up with 594 unique items in an evaluation set that are predicted confidently by both RFAA and RF2. For each item, we make 3 predictions from both RFAA and RF2 to remove any artifacts that come from random seed, and pick the prediction that either minimizes inter-chain PAE or maximizes pLDDT, respectively.

We measure the all-atom RMSD of the predicted protein structure relative to the crystal after kabsch alignment on the backbone atoms. We also measure the *ligand pocket* RMSD, where the ligand pocket is defined as the set of residues that have at least one atom within 5Å of the ligand in the crystal structure. The ligand pocket RMSD is then computed as the all-atom RMSD of said residues after kabsch alignment on the backbone atoms of the same residues.

We observe a statistically significant difference (paired t-test) between the RMSD values from

RFAA and RF2 (Fig. S2.5A). We depict several examples of structures that RFAA predicts better than RF2, likely due to the added ligand context (Fig. S2.5B-D). One illustrative example is the PDB entry 7rjj in our test set. For this target, RFAA generates an accurate prediction while RF2 predicts an incorrect, "open" conformation of a helix forming the ligand-binding pocket (Fig. S2.5D, pink structure). This open conformation is present in the most sequence-similar example in the training set (Fig. S2.5D, yellow structure). However, a different example in the training set, which has lower sequence similarity to 7rjj, has the helix in a "closed" conformation in the presence of a ligand (Fig. S2.5D, orange structure). We hypothesize that RFAA uses the presence of the ligand to better disambiguate alternate conformations of similar proteins seen during training.

Predicting Structures With and Without Ligands A natural question that arises is whether or not RFAA can predict conformational shifts in a protein with and without a ligand partner present. For those same set of 594 items that are confidently predicted by RFAA and RF2, we make predictions of the protein structure without the added ligand context using the RFAA model (best of three predictions by PLDDT) and plot the binding site RMSD between predictions with and without the ligand (Fig. S2.5E). We observe some differences between predictions made with and without ligand partner present - in particular, RFAA is capable of predicting conformational shifts for which both apo and holo states are well-represented in its training set. However, we note that it is difficult in general to evaluate how well RFAA changes to a protein upon binding and do not expect the model to generalize to completely novel conformational changes. We expect that future work will make intentional train/test splits that exclude specific conformational changes to assess whether the network can generalize to novel conformational shifts but consider that outside the scope of the work presented in this manuscript.

CAMEO Baseline Servers

In Fig. S2.4A-B, we present a per-ligand breakdown of the targets predicted in the CAMEO challenge by both the RFAA server and the CAMEO baselines based on classical docking methods from weeks 08/12/2023 to 09/02/2023. The CAMEO baselines first predict the

structure of the protein chain(s) using the SWISS-MODEL software and then dock the ligands using AutoDock Vina sequentially with a bounding box that covers the volume of the entire protein [55, 124]. The only difference between between the “Vina” and “AD4” baselines is the scoring function used when running AutoDock Vina (vina or AutoDock 4).

We show some common failure modes of the network in Fig. S2.4D. The most common failure mode is placing the molecule in a different orientation than the crystallized pose. These orientations usually also have shape and chemical complementarity to the protein structure but there is another potentially "lower energy" dock that is not sampled. Other failure modes we noticed in CAMEO is that the network predicts regions that are unresolved in the true structure which change the geometry of the binding pocket and thus change the dock of the small molecule. The final common failure mode is choosing the predict more buried positions for complexes that interact on the protein surface.

Prediction of Metal Ions

The set of protein-metal ion complexes constructed in subsection 2.10.7 were predicted. To analyze the results of this evaluation, we restricted the dataset to 6 transition metals that are known to have coordinated binding sites (Zn, Mn, Fe, Cu, Co, Ni). We find that RFAA makes accurate predictions for these metals (Fig. S2.3). Similar to the other datasets, we find that the predicted alignment error (all protein residue frames vs metal ion since metal ion does not have a canonical frame) accurately correlates with accuracy (Fig. S2.3).

Chapter 3

ATOM LEVEL ENZYME ACTIVE SITE SCAFFOLDING WITH RFDIFFUSION2

This chapter is published as Ahern, W.†, Yim, J.†, Tischer, D.†, Salike, S., Woodbury, S., Kim, D., Kalvet, I., Kipnis, Y., Coventry, B., Altae-Tran, H., Bauer, M., Barzilay, R., Jaakkola, T., Krishna, R.*, Baker, D.* Atom level enzyme active site scaffolding using RFdiffusion2. bioRxiv. Preprint. 2025.

W.A., D.T. and D.B. conceived the study. W.A., J.Y., D.T., S.S., and H.R.A.T contributed to development of RFdiffusion2. W.A. trained all versions of RFdiffusion2 described in the study. W.A., S.S., and R.K. performed *in silico* benchmarking. W.A., S.S., J.Y., and R.K. analyzed RFdiffusion2 results. S.S., S.M.W., D.K., I.K., Y.K., and B.C. experimentally characterized designs. W.A, J.Y., R.K., and D.B were responsible for the manuscript writing. S.S., S.M.W., D.K., and I.K. were responsible for writing *in vitro* details. D.B., R.K., T.S.J. and R.B. offered supervision throughout the project.

3.1 Introduction

A grand challenge in *de novo* protein design is the generation of enzymes that catalyze novel reactions. *De novo* enzyme design starts from a detailed description of an active site composition and geometry predicted to catalyze the reaction of interest. This active site description, called a theozyme, describes placement of protein functional groups around the reaction transition state(s) and any reaction cofactors [64, 86]. The *de novo* enzyme design task is to generate protein scaffolds that accommodate such theozymes. Pre-deep learning methods such as RosettaMatch searched through sets of already existing native or designed [142] scaffolds for possible placements of the catalytic residues. While many enzymes were designed using this approach, it was restricted to theozyme geometries that

could be matched to the input scaffold set [146]. Advances in deep learning with diffusion models [6, 61, 84, 112] have removed the need for scaffold libraries for many substructure scaffolding tasks by directly sampling diverse proteins containing the desired substructure (motif) through a technique known as motif scaffolding [27, 68, 82, 132]. However, thus far these methods all operate on a backbone level representation of proteins as a series of amino acid residues, and consequently, can only scaffold motifs represented at the backbone level.

Current approaches attempt to overcome this limitation for atom level active site descriptions by enumerating possible conformations and sequence indices for the catalytic residues and then in a separate step using motif scaffolding to generate proteins which scaffold these backbone positions [23, 77]. While active enzymes have been generated using this approach, it is computationally inefficient and does not scale to more complex active sites as the number of combinations of backbone coordinates to scaffold grows exponentially with the number of catalytic residues. A method capable of scaffolding complex theozymes described at the atom level would have wide-spread applications for enzyme design and beyond [24, 65, 107].

We reasoned that substantially improved performance on more complex enzyme active site scaffolding challenges could be achieved by a generative model capable of selecting the conformations and sequence indices of the catalytic residues by modeling the full joint distribution of rotamers, sequence indices, and scaffolds conditioned directly on atom level active site descriptions. With RFdiffusion2, we set out to extend RosettaFold diffusion All-Atom (RFdiffusionAA) [76] to generate structures conditioned on these minimal active site descriptions.

3.2 Atomic Motif Conditioning

To address this challenge, we sought to generalize motif scaffolding beyond sequence indexed, backbone level motifs. Prior motif scaffolding methods represent motifs as “backbone frames", in which each amino acid’s N, Cα, C backbone atoms are parameterized as an element in SE(3). Each motif frame requires a pre-specified sequence index that indicates the frame’s location along the backbone chain. In contrast, the protein component of an atom level active site description includes only the side chain functional group atoms, not the backbone, of the

participating amino acid residues; these residues can be anywhere along the sequence, i.e. of unknown index. While the indexed frame representation is sufficient for scaffolding large contiguous domains that can be accurately described at the residue level, it is insufficient to express the task of scaffolding disconnected groups of atoms belonging to residues of unknown indices.

Our approach is to create an extended representation with differing levels of resolution and index information that is capable of expressing more complex motifs. In RFdiffusion2, we use the RosettaFold All-Atom neural network architecture in which each residue in the input and output can be represented as a frame or as heavy atom coordinates, i.e. an atomized residue [76]. During training, we represent some residues with frames and atomize others. For each atomized residue, the network learns to model the distribution of side chain poses. By providing known coordinates for some side chain atoms, which we term the atomic motif, the network learns to model the distribution of proteins conditioned on the inclusion of such atomic substructures. At inference time, we can then condition on the coordinates of individual protein atoms such as the side chain (or backbone) functional groups present in a theozyme. This ability to condition on individual atoms rather than entire residues allows us to forgo inverse rotamer sampling [146] used in previous approaches and instead allow the model to simultaneously infer an appropriate rotamer and scaffold.

We can further extend the representation to remove the need to know the sequence indices of a motif in order to scaffold it. During training, we select subsets of residues, duplicate them, and remove their index features to create *unindexed residues*. Without any auxiliary losses, the network learns that unindexed residues are always superimposed on indexed residues in unnoised structures. By providing coordinates for unindexed residues, which we term an *unindexed motif*, the network learns to model the distribution of proteins conditioned on the inclusion of a known substructure at unknown indices. At inference time, we then have the flexibility to condition on a motif consisting of residues with known or unknown indices because theozymes do not prescribe the sequence indices of their constituent residues. The ability to condition on motifs without specifying their indices enables us to forgo naive index sampling used in previous approaches and instead allow the model to simultaneously infer

indices for the motif while scaffolding it (Figure 3.2B).

The resulting model, RFdiffusion2, can generate proteins conditioned on a broad range of motifs including side chain motifs, motifs without known sequence indices, and ligand motifs. When training RFdiffusion and RFDiffusionAA, we found performance started to worsen over extended periods of training. Through a combination of auxiliary losses and self-conditioning [32], these methods were able to achieve high success rates on their respective tasks in short training sessions. Due to the complexity of the unindexed atomic motif scaffolding task, we expected that a stable objective would be required. To this end, we train RFdiffusion2 with flow matching [6, 84], a simpler framework for diffusion models [45, 88] shown empirically to have improved training and generation efficiency in other domains [85]. Briefly, this framework interpolates a training example towards a noise sample and trains a neural network to denoise by predicting the original, uncorrupted example. If trained to denoise with sufficient accuracy, the model can sample from the data distribution by iteratively denoising a sample drawn from the noise prior. Our representation of the data distribution contains both atoms and backbone frames which are elements of $\mathbb{R}^3$ and SE(3), respectively. Flow matching on $\mathbb{R}^3$ follows its original derivation using Gaussian probability paths while for SE(3) we follow the formulation in FrameFlow [143, 144] that utilizes Riemannian flow matching [31] and removes approximations for rotational losses present in the RFdiffusion [132]. With these improvements, RFdiffusion2 trained stably from randomly initialized neural network weights, does not require auxiliary losses, or use self-conditioning. Decoupling RFdiffusion2 from structure prediction is an important step to remove constraints around the neural network architecture as well as enable new generative modeling tasks. Our training dataset consists of biomolecular structures from the Protein Data Bank (PDB) [17] that include proteins, protein–small molecule complexes, protein-metal complexes, and covalently modified proteins, filtering out common solvents and crystallization additives [76]. Each structure undergoes a motif extraction procedure to construct a motif-scaffolding training example. First, we select a random subset of residues to be the motif. Motif residues are chosen uniformly at random to ensure RFdiffusion2 sees a diverse range of interatomic geometries in the motif. Second, each motif residue is represented either as a frame or atomized residue. If a motif residue is

atomized, only a random subgraph of the amino acid heavy atoms are provided as the motif. Third, we decide whether the motif will be featurized as an indexed or unindexed. Lastly, for any ligand present, we sample a random connected subgraph of its heavy atoms to be provided as a motif with Relative Accessible Surface Area (RASA) [121] labels for each atom in the ligand intermittently provided. We resample the motif on each training step as a form of data augmentation to ensure that the same noised structure is not always shown alongside the same motif. We train the final model for 17 days using 24 A100 Nvidia GPUs.

In our early experiments with flow matching, we found that the choice of how the ground truth structures were centered relative to the origin in training significantly affected the quality of inference outputs. A natural strategy is to globally center each ground truth structure; however, this leaks the offset between the scaffold center of mass and the motif. At inference time, this strategy requires the exact specification of the desired offset which is usually not known. A common fix for this pathology in motif-scaffolding diffusion models is to center ground truth structures on the motif, allowing the model to determine the offset between the scaffold center of mass and the motif [68, 144]. However even when the motif is centered, due to the interpolation scheme used in flow matching, the model is able to exactly determine this offset from any partially noised structure. The resultant behavior is that in the first denoising step, in which the network receives pure noise, it predicts an offset that it does not learn to refine in subsequent denoising steps. We instead introduce *stochastic centering* which first center the ground truth structure and add a small global translation sampled from a 3D Gaussian such that the noised input structures only encode an approximate offset between the motif and scaffold. This enables the model to refine the placement of the motif within the overall structure over the course of the inference trajectory. At inference time, users supply a prior belief about the placement of the motif through a special `ORI` pseudo-atom that specifies the approximate center of mass of the generated structure. This enables enzyme designers to control the active site and transition state orientation relative to the protein core (Figure 3.2D). For example, given an elongated small molecule or transition state with one end quite polar or charged, placement of the `ORI` token adjacent to the opposite end (or displaced from this end along a vector running through

the long axis of the molecule) results in a designed binder or enzyme with a binding pocket extending radially from the center of the protein with the polar/charged end of the small molecule exposed to solvent.

RFdiffusion2 provides two additional conditioning capabilities of the ligand that are useful in *de novo* enzyme design. First, to provide finer control over the depth at which each reactant and/or cofactor is buried within the protein, we enable users to specify the RASA of each atom. By providing the RASA of each ligand atom 50% of the time during training, RFdiffusion2 learns to generate structures that respect those atomwise conditions at inference time when provided (Figure 3.2C). Second, the user may know the ligand atoms of a transition state but may not know the full ligand conformer. We allow the user to specify "partial ligands" where only the known ligand atoms are provided while RFdiffusion2 infers the rest of the ligand conformer. A brief analysis on the physical plausibility of the generated conformers shows they match closely with RDkit generated conformers. We find RFdiffusion2 can respect partial ligands by sampling physically realistic conformers (Figure 3.2D), removing the need to naively resolve the ligand conformer with an external tool *a priori*. Together, the conditioning capabilities opens up greater control over the geometric properties of the protein-ligand complex during inference.

3.3 Representations

Amino acid representation. RFdiffusion2 introduces a novel biomolecule representation that can capture a wide range of inputs and outputs. Amino acid residues can be represented as follows.

- $X = [x_1, \ldots, x_N]$ is the collection of N *indexed* amino acids residues x_i with subscripts $i \in [1, \ldots, N]$ describing where each residue is along the polypeptide chain via a mapping `idx` (described later).

- $\bar{X} = [\bar{x}_1, \ldots, \bar{x}_{\bar{N}}]$ is the collection of $\bar{N}$ *unindexed* amino acid residues $\bar{x}_i$ with subscripts $i \in [1, \ldots, \bar{N}]$ that *do not* correspond to a location along the polypeptide chain. We use indices i to enumerate through unindexed residues, but they have no connection to

the polypeptide chain.

Each residue x (or $\bar{x}$) can be represented as a frame $x \in \mathrm{SE}(3)$ or atomized $x \in \mathbb{R}^{14\times 3}$ following the amino acid 14 atom representation [72]. We acknowledge the abuse of notation since $x_i^{(t)}$ and $\bar{x}_{\bar{i}}^{(t)}$ have different dimensionalities and different spaces. We treat X (and $\bar{X}$) as a "jagged" tensor[1] where each row of a dimension can have a different length to avoid unnecessarily verbosity, i.e. $\mathrm{Shape}(x_i^{(t)}) \neq \mathrm{Shape}(x_j^{(t)})$ for some $i \neq j$. This notation helps simplify the exposition. We will use the over head bar notation to refer to unindexed variables. For example, $\bar{X}$ is the collection of unindexed residues, $\bar{N}$ is the number of unindexed residues. Providing the sequence index as an input feature helps the model form a polypeptide chain by placing x_i in between its neighboring residues in the chain. Unindexed residues are a way of prompting RFdiffusion2 to figure out the sequence index placement for $\bar{x}_i$.

Before we describe how unindexed residues work, we must describe how motifs are represented. There are two types of motif residues, indexed and unindexed, which we denote as

$$X^M = \{x_i^M : i \in \mathcal{M}\}, \qquad \bar{X}^M = \{\bar{x}_i^M : i \in \mathcal{M}\} \tag{3.1}$$

where $\mathcal{M} \subset [1, \ldots, N]$ is the collection of subscripts for indexed residues in the motif. Typically in motif-scaffolding with generative models, the model is provided X^M and is tasked with generating the scaffold X^S defined as the remaining residues in X not provided for in X^M

$$X^S = \{x_i : i \in [1, \ldots, N]/\mathcal{M}\} \tag{3.2}$$

However, the scaffold cannot be defined for motif residues in $\bar{X}^M$ which do not have known sequence indices. This can be overcome by guessing the sequence indices for each unindexed motif residue, but the trial and error for guessing sequence indices in a combinatorial space of $[1, \ldots, N]$ is infeasible to enumerate. If a generated enzyme is bad, how can we know if it is due to the generative model or due to a poor choice of sequence indices?

---

[1]Pytorch introduced jagged tensors for handling elements of different lengths. See `https://pytorch.org/FBGEMM/fbgemm_gpu-overview/jagged-tensor-ops/JaggedTensorOps.html`

Unindexed residue satisfaction. Our representation allows for a solution where RFdiffusion2 can handle $\bar{X}$ without manually guessing sequence indices at the start. At a high level, we set up an input representation where the model generates a full protein X but with the constraint that each unindexed residue in $\bar{X}$ has to have an overlapping indexed residue. Let us assume $X^M = \emptyset$ to simplify our notation and that $\bar{x}_i \in \text{SE}(3)$ for each $i \in [1, \ldots, \bar{N}]$. A unindexed residue $\bar{x}_i$ is defined as being *satisfied* if the following holds

$$\left(\min_{j \in [1,\ldots,N]} \|\text{Log}_{\bar{x}_i}(\bar{x}_i) - \text{Log}_{x_j}(x_j)\|_g \right) < \epsilon \tag{3.3}$$

for some very small epsilon ϵ and where $\|\cdot\|_g$ is the norm with the Riemannian metric g for SE(3) and Log_x is the logarithmic map onto the tangent space centered at point $x \in \text{SE}(3)$. If $\bar{x}$ is satisfied then we can extract the sequence index that RFdiffusion2 assigned with the following

$$\arg\min_{j \in [1,\ldots,N]} \|\text{Log}_{\bar{x}_i}(\bar{x}_i) - \text{Log}_{x_j}(x_j)\|_g \tag{3.4}$$

RFdiffusion2's must satisfy each $\bar{x}_i$ with a indexed residue. In other words, it is protein generation while adhering to constraint satisfaction that we specify (or prompt) through our representation.

Partially defined atomized motif residues. We have discussed $\bar{x}_i^M$ as if the residue coordinates are fully specified. However, it is often the case only the side chain functional groups are known for $\bar{x}_i^M$. Here, $\bar{x}_i^M$ is represented as an atomized residue, i.e. in $\bar{x}_i^M \in \mathbb{R}^{14\times 3}$. There are two tasks RFdiffusion2 must perform for partially specified unindexed atomized residues. First, it must satisfy the Eq. (3.1) constraint in finding a indexed residue to satisfy $\bar{x}_i^M$. Second, it must generate the unspecified atoms in $\bar{x}_i^M$. We will denote $\bar{x}_i$ as the atom coordinates in $\bar{x}_i^M$ that are to be generated by RFdiffusion2. $\bar{x}$ takes the place of inverse rotamer sampling that was previously used when motif residue atom coordiantes were not fully specified. We now tasks the model with generating the atom coordinates that become the rotamers in the motif residue. Partially specified atomized residues are not unique to unindexed motif residues. Indeed, x_i^M can also be specified as atomized residues with missing atom coordinates that must be inferred by the model.

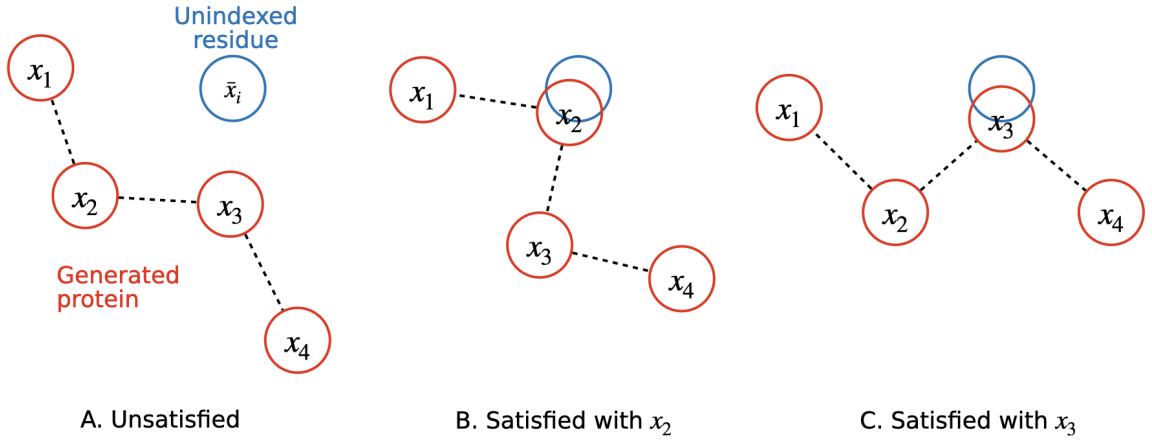


Figure 3.1: Diagram of unindexed residue satisfaction. In red are four indexed residues that make up the generated protein while in blue we show a single unindexed motif residue $\bar{x}$. **A.** The unindexed residue is not satisfied since none of the residues in the generated protein overlap with it. **B.** Here it is satisfied since x_2 overlap. The generated protein has placed this particular motif residue at index 2. **C.** Again it is satisfied but this time with x_3. Whichever residue in the generated protein used to satisfy $\bar{x}$ does not matter. It is up to the model to select which indexed residue to use.

Ligand representation. Following RFdiffusionAA, we represent ligands, denoted $L \in \mathbb{R}^{K\times 3}$, as a collection of K 3D atomic coordinates. Similar to partially defined atomized residues, ligands can be partially defined with the unknown atom coordinates left to RFdiffusion2 to infer. We use L^M to denote the ligand motif that holds the unnoised ligand atoms provided to the model.

3.4 Model Architecture

The neural network architecture used in RFdiffusion2 closely resembles RosettaFold All-Atom (RFAA) and RFdiffusion All-Atom (RFdiffusionAA) [76]. We refer to those works for more details while providing a brief description here and the changes we made.

RFdiffusion2 follows the three tracks introduced in RosettaFold (RF) [10] where the neural network can accept 1D sequence information, 2D distance and bond information and 3D coordinate information. While we train the network from scratch, we choose to maintain the input tensor shapes of RFAA because the network architecture was optimized for the

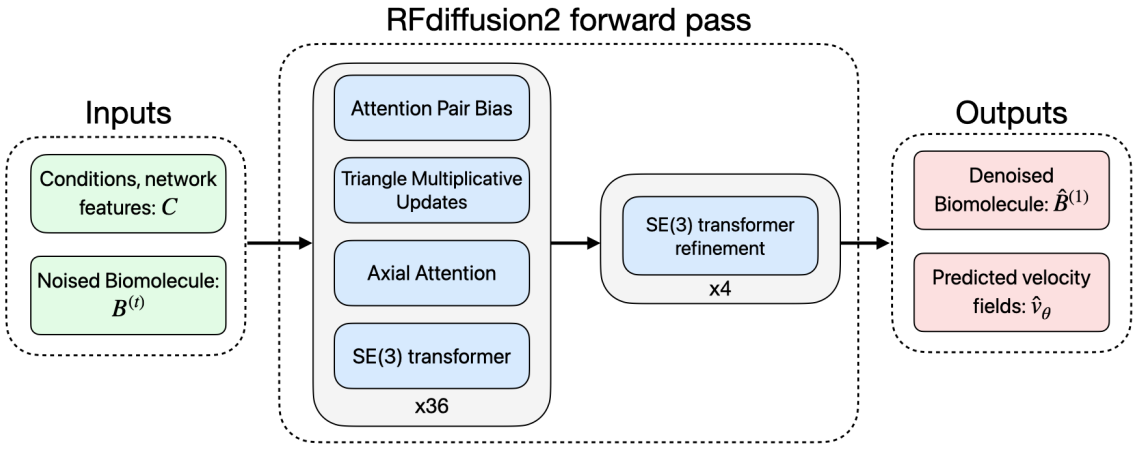


Figure 3.2: Diagram showing the neural network architecture of RFdiffusion2. The middle of the diagram labeled RFdiffusion2 forward pass shows the neural network operations including Attention Pair Bias, Tri. Mult. Update, Axial Attention, SE(3) transformer, and SE(3) transformer refinement (see [76] for details on each operation). We use plate notation to show the number of times each operation is performed. The outputs include the denoised biomolecule that is converted into the vector fields used in the flow matching loss.

structure prediction inputs. Briefly, RFAA introduced the ability to model arbitrary small molecules with the following innovations:

- Adding tokens for all atoms found in small molecules in the PDB.
- Adding explicit bond feature inputs
- Adding chirality inputs which break 3D chiral symmetries
- Generalizing the positional encoding to break permutation symmetry for polymers but maintain the symmetry for small molecules
- Scaling the trunk attention heads and channels to improve performance
- Generalizing the structure module to operate on frames for polymer residues and global translations for ligand atoms

Architectural Improvements We largely followed the RFAA architecture [76]. In training RFAA for structure prediction, we found there was training instability associated with the rotation prediction which has also been noted in AF2 [72]. We used the same heuristic solution that was used in AF2 where the gradients were stopped between rotation predictions in the network. This was exacerbated since in RFAA we were predicting a structure at every block in the network instead of the structure module at the end of the network as in AF2. In RFdiffusion2, we found that allowing the gradients to flow through rotations of the SE(3) Transformer blocks in the trunk of the network did not cause instability. We still stop the rotation gradients in the refinement blocks at the end of the network.

Following RFdiffusionAA, we zero out the rotation and translation updates for motif residues and atoms. In this case, since the motif is provided within the noisy coordinates, the motif coordinates will never move during the trajectory.

RFAA performs recycling of latent features as introduced in [72]. In RFdiffusion2, we do not use the recycling features and they are always initialized to zeros. This speeds up running the model.

3.5 Discussion

RFdiffusion2 removes expert intuition necessary with prior backbone motif scaffolding and scaffold library methods, and can design enzymes with *in vitro* catalytic activity. RFdiffusion2 enables direct scaffolding of ideal active sites described at the atom level without pre-specifying sequence indices or enumerating side chain rotamers.

There are several avenues for improvement. Despite RFdiffusion2's success in obtaining active enzymes across four reactions, the enzymes designed by RFdiffusion2 are not as active as native enzymes. Our theozymes might not be capturing all the necessary interactions for high activity and RFdiffusion2 might be able to sample higher activity enzymes by expanding our theozyme definition to include more interactions that are necessary for catalysis [77]. Automating the design of enzymes from theozymes opens up for the first time the possibility of large-scale testing of varied theozymes to broaden our understanding of enzymes and

validate mechanistic hypotheses of much wider scope than those testable with catalytic residue knockout experiments or directed evolution. Alternative neural network architectures such as Diffusion Transformers [97] and modules from AlphaFold3 [2] for all-atom tasks could improve RFdiffusion2 which uses the neural network architecture of RFAA. Finally, we expect that co-designing the protein sequence [27] and side chains [33] outside the active site could lead to more favorable pocket interactions with the substrate and potentially enable sequence-based guidance based on experimental kinetics data.

RFdiffusion2 should be immediately useful to protein designers working on design problems requiring atomic resolution modeling such as small molecule binding and enzyme design. We expect that the introduction of RFdiffusion2 and the AME benchmark will open up new research efforts in the machine learning community exploring designing new modeling approaches for atomic resolution protein design. To this end, we are making the RFdiffusion2 inference and training code freely available to the research community.

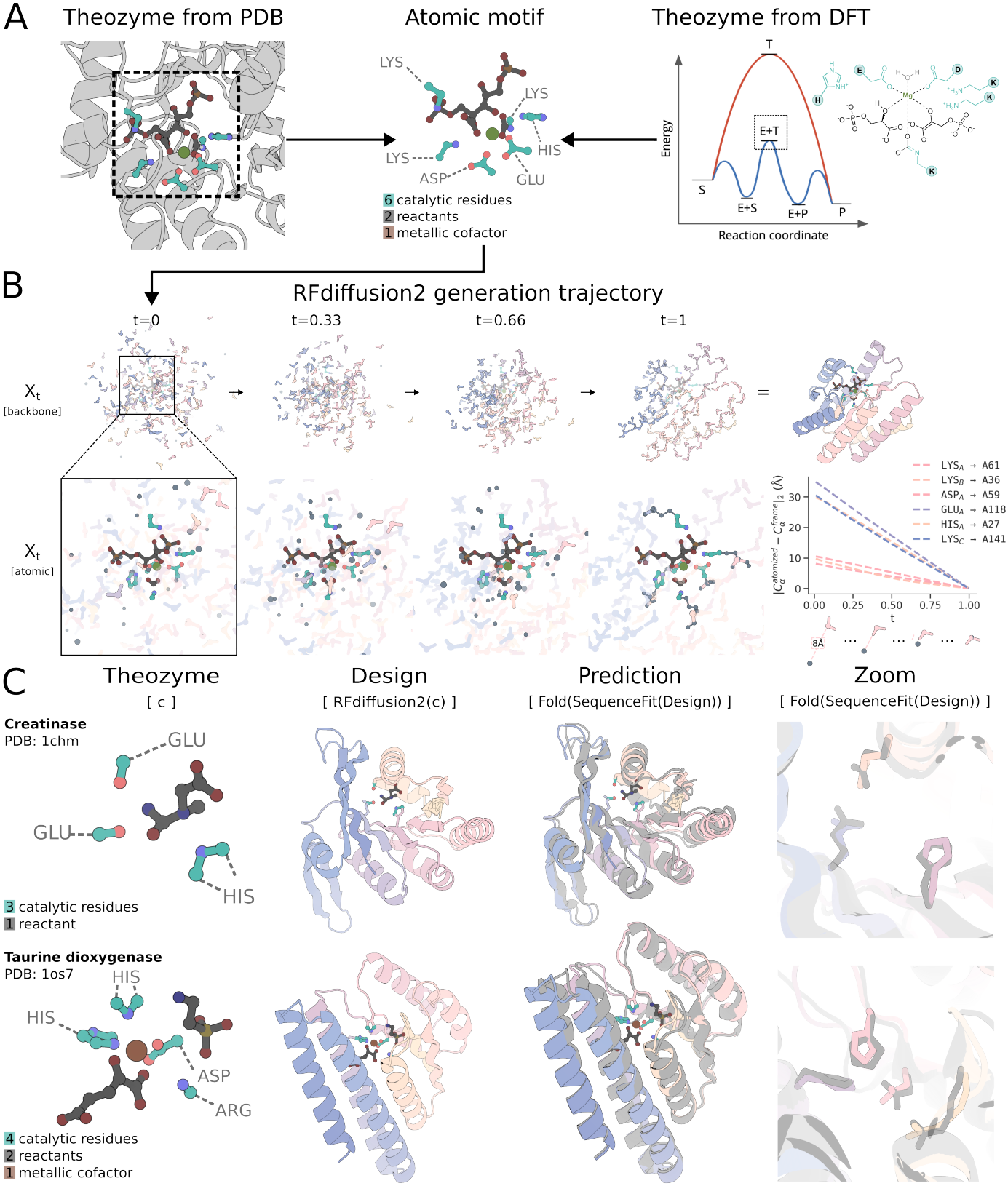
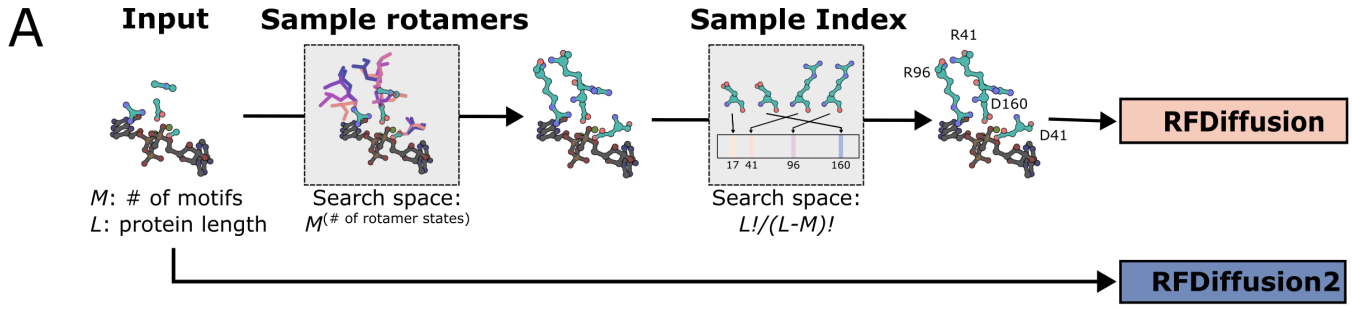
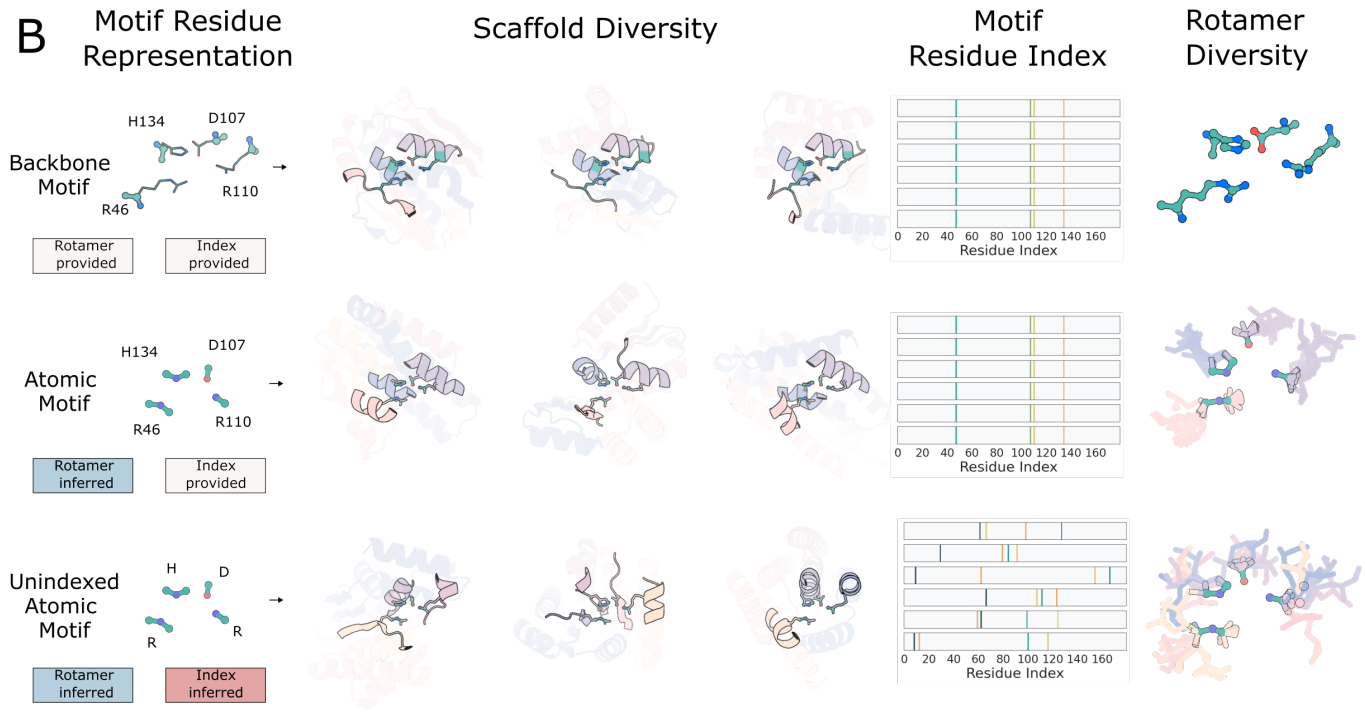
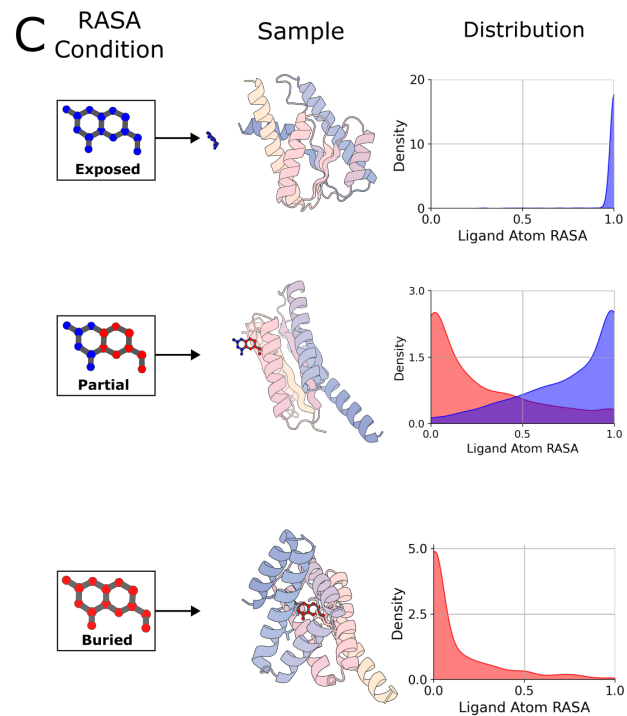
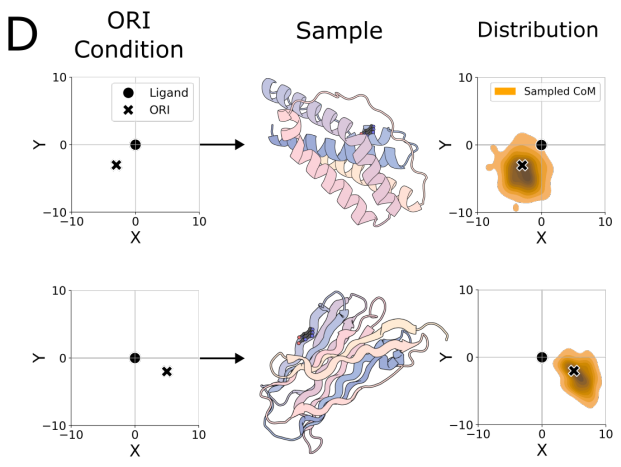

A
Theozyme from PDB
Atomic motif
Theozyme from DFT
LYS
LYS
LYS
HIS
ASP
GLU
6 catalytic residues
2 reactants
1 metallic cofactor
T
E+T
Energy
S
E+S
E+P
P
Reaction coordinate
B
RFdiffusion2 generation trajectory
t=0
t=0.33
t=0.66
t=1
X_t [backbone]
X_t [atomic]
$LYS_A \rightarrow A61$
$LYS_B \rightarrow A36$
$ASP_A \rightarrow A59$
$GLU_A \rightarrow A118$
$HIS_A \rightarrow A27$
$LYS_C \rightarrow A141$
$|C_\alpha^{atomized} - C_\alpha^{frame}|_2$ (Å)
t
8Å
C
Theozyme
[c]
Design
[RFdiffusion2(c)]
Prediction
[Fold(SequenceFit(Design))]
Zoom
[Fold(SequenceFit(Design))]
Creatinase
PDB: 1chm
GLU
GLU
HIS
3 catalytic residues
1 reactant
Taurine dioxygenase
PDB: 1os7
HIS
HIS
ASP
ARG
4 catalytic residues
2 reactants
1 metallic cofactor

Figure 3.2 *(previous page)*: RFdiffusion2 overview. **A.** *De novo* enzyme design starts from a configuration of catalytic groups around the reaction transition state(s) (a theozyme) generated using quantum chemistry, protein structural analysis and/or chemical reasoning. **B.** RFdiffusion2 generates protein structures that support the theozyme. Row 1: The backbone trajectory shows the amino acid residue frames (pastel) as they transform from a sample drawn from the noise distribution into a protein backbone. Row 2: Zoom in of Row 1, showing the non-motif side chain atoms (slate grey) connecting the atomic motif (teal) with the protein backbone. At $t = 1$ the intra-residue bonds are shown for the atomized residues. (right) The distances between the Cα coordinates of the unindexed, atomized residues and the backbone residues they superimpose at $t = 1$. Over the course of the trajectory, the model matches these unindexed residues to indexed residues of the protein backbone, such that by the end of the trajectory the unindexed residue's Cα occupies the same location as the Cα of the protein backbone in Euclidean space. **C.** The design pipeline starts from the input theozyme, followed by RFdiffusion2 to generate the structure, and LigandMPNN to generate amino acid sequences that encode the structure and stabilize the transition state. Designs are evaluated by all atom structure prediction (using Chai-1, AF3, etc) and are considered successful if the design (pastel) and prediction (light gray) align to a sufficient degree. Two representative examples of consistency between design model and predicted structure at the level we take to constitute a success are shown in the right panels. The two cases pictured are the Creatinase and Taurine dioxygenase motifs from the AME benchmark. (AME IDs: M0096_1chm, M0129_1os7).


A
Input
Sample rotamers
Sample Index
R41
R96
D160
D41
RFDiffusion
M: # of motifs
L: protein length
Search space:
M(# of rotamer states)
17 41 96 160
Search space:
L!/(L-M)!
RFDiffusion2


B
Motif Residue Representation
Scaffold Diversity
Motif Residue Index
Rotamer Diversity
Backbone Motif
H134
D107
R46
R110
Rotamer provided
Index provided
Atomic Motif
Rotamer inferred
Index provided
Unindexed Atomic Motif
H
D
R
Rotamer inferred
Index inferred
0 20 40 60 80 100 120 140 160
Residue Index


C
RASA Condition
Sample
Distribution
Exposed
Partial
Buried
Density
Ligand Atom RASA


D
ORI Condition
Sample
Distribution
Ligand
ORI
Sampled CoM
X
Y

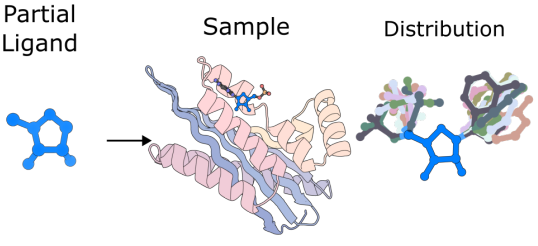

E
Partial Ligand
Sample
Distribution

Figure 3.2 *(previous page)*: Motif scaffolding with RFdiffusion2. **A.** In the original RFdiffusion, two preprocessing steps are required to transform an unindexed atomic motif into a suitable input. These steps – inverse rotamer sampling and sequence index sampling – both require selecting from an exponentially large search spaces of $L!/(L-M)!$ and $M^{(\#\text{ of possible rotamer states})}$ respectively, where L is the number of residues while M is the number of residues that are needed for the active site. RFdiffusion2 does not require such preprocessing steps and can scaffold unindexed atomic motifs directly. **B.** RFdiffusion2 can be conditioned on motifs in different representations. Three versions of the same motif (M0904, PDB 1qgx) from AME) are shown on the left most column, different backbone samples in the middle columns, and the resulting diversity of sequence indices and rotamers on the right most columns. (i) The backbone motif includes a pre-specified rotamer and index as required by RFdiffusion. (ii) The atomic motif has pre-specified sequence indices but unspecified side chain conformation. (iii) Only unindexed atom positions are provided, not the residue indices or side chain rotamer conformations. The rotamer and sequence indices are sampled during the RFdiffusion2 trajectories, increasing the diversity of possible solutions to the motif scaffolding problem. **C.** Each ligand atom can be labeled with a RASA category to control how solvent exposed the ligand is. The example RASA conditions are in the left column, a backbone sample with the ligand in the middle, and the distribution of ligand atom RASA from 100 designs with the RASA condition. When all atoms are labeled as exposed, the ligand RASA is concentrated around 1.0 and the backbone does not come into contact with the ligand. Conversely, when all atoms are labeled as buried, the ligand RASA is concentrated around 0.0; the sample shows the backbone almost completely covering the ligand. Labeling half the ligand as exposed and the other half as buried leads to RFdiffusion2 generating backbones that only bind to the buried side of the ligand. **D.** RFdiffusion2 can be provided with an `ORI` token that specifies the desired center of mass (CoM) of the scaffold with respect to the ligand. Two different `ORI` positions are shown in the left column. The middle column shows samples with scaffolds centered at the indicated `ORI` token positions. The distribution of CoMs from 100 sampled designs with the `ORI` token show that the scaffolds generally follow the `ORI` condition of where to place the scaffold. **E.** RFdiffusion2 can be provided with partial ligand input in which case it must sample the remaining ligand degrees of freedom while generating the protein. The left column shows the partial ligand input. The middle column shows in gray a conformer along with the protein generated by RFdiffusion2. Finally, the right column shows the distribution of 10 generated conformers.

Chapter 4

DEMOCRATIZING NEURAL NETWORK DEVELOPMENT FOR BIOMOLECULAR MODELING AND DESIGN

This chapter contains unpublished work that includes contributions from the following authors: Frank DiMaio, Nate Corley, Jason Yim, Simon Mathis, Tuscan Thompson, Woody Ahern, Magnus Bauer and David Baker.

4.1 Introduction

In Chaper 2 and 3, we describe the development of neural networks trained on datasets of biomolecular assemblies in the PDB. In this chapter, we describe a general framework for future development of neural networks for biomolecular assembly prediction and design. The hallmark of this framework is the use of a single software package to train and evaluate several state-of-the-art models. This framework, called ModelHub, allows for rapid development and testing of new models and architectures. The development of ModelHub is an important step towards the goal of creating a general-purpose neural network for biomolecular assembly prediction and design. In the following sections, we describe the strength of the ModelHub framework and the results of using it to train and evaluate several models.

4.2 Model Ensembles

We reasoned that a framework for training multiple models simulataneouly would enable parallel investigation of the design space of neural network models for biomolecular structure prediction. Using this framework, we trained 4 models: 1) the model presented in [76], 2) a model with an updated "recycling" procedure which does not recycle side chain torsion angles between forward passes of the network, 3) similar to 2 but 48 network blocks instead of 36, and 4) similar to 3 but applying a loss on the predicted rotations of atom nodes using

Table 4.1: Model Performance on Posebusters

Model	% <2 angstroms (Top 1)	% <2 angstroms (Top 5)
Paper model	36	NA
No sidechains recycled	33	NA
48 layers	33	NA
Atom rotation loss	23	NA
Ensemble	NA	48

the heuristic atom frame assignment procedure proposed in [76]. We find that none of these models are more accurate than our published model but that they predict different cases in the PoseBusters set [25] accurately.Additionally, we find in many cases the predicted confidence of the most accurate pose is accurately reflected by the model's "PAE Interaction" confidence metric. When ensembling these 4 models, we find a Top 5 accuracy of 48 % <2Åand 66% <3Å(Table 4.1).

4.3 Training Generative Model for Structure Prediction

In the previous section, we showed that a large density of predicted structures fall into the 2-4 Åerror range. We hypothesized that this was due to the model mean-seeking over multiple possible structures and producing slightly inaccurate averages of possible structures. We decided to test this hypothesis by training a generative model that could sample from the distribution of possible structures. We largely follow the formalisms used in chapter 3, which we enumerate below.

Background

Flow matching [6, 84] is a method for training Continuous Normalizing Flows (CNFs) which describe a process in which samples from a prior distribution $x^{(0)} \sim p^{(0)}$ are transformed into a sample from the data distribution $x^{(1)} \sim p^{(1)}$ via a *marginal flow*, $\phi^{(t)}$, defined in terms of

a *marginal vector field* $v^{(t)}$ as

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi^{(t)}(x) = v^{(t)}(\phi^{(t)}(x)) \quad \text{where} \quad \phi^{(0)}(x) = x \tag{4.1}$$

for some time $t \in [0, 1)$. We can derive the exact form of the flow or vector field with integration and differentiation of the Ordinary Differential Equation (ODE) in Eq. (4.1) if the closed form of one is known. Unfortunately, neither the marginal vector field nor flow are available in closed form but the key idea of FM is to *learn* the marginal vector field by regressing the *conditional* vector field $u(x^{(t)}|x^{(1)}) = \frac{\mathrm{d}}{\mathrm{d}t}x^{(t)}$ from which we can integrate to derive the *conditional* flow $x^{(t)} = \phi^{(t)}(x^{(0)}|x^{(1)})$. The closed form of $x^{(t)}$ depends on the manifold which the data is generated on. In general, we can write the conditional flow $x^{(t)}$ as a geodesic path

$$x^{(t)} = \mathrm{Exp}_{x^{(0)}}\left(t \cdot \mathrm{Log}_{x^{(0)}}(x^{(1)})\right) \tag{4.2}$$

where $\mathrm{Exp}_{x^{(0)}}$ and $\mathrm{Log}_{x^{(0)}}$ are the exponential and logarithmic maps at point $x^{(0)}$. Then the conditional vector field can be written as

$$u(x^{(t)}|x^{(1)}) = \frac{\mathrm{Log}_{x^{(t)}}(x^{(1)})}{1-t}. \tag{4.3}$$

The FM objective is then

$$\mathbb{E}_{\mathcal{U}(t;0,1),p^{(1)}(x^{(1)}),p^{(0)}(x^{(0)})}\left[\|u(x^{(t)}|x^{(1)}) - \hat{v}(x^{(t)})\|_g^2\right] \tag{4.4}$$

where $\hat{v}$ is the output of a neural network and g is the Riemannian metric for the Manifold on which x lies. Since we only deal with Euclidean and Lie groups, we can assume the norm takes the form of a L2. Using Eq. (4.3), we can reparameterize the model output in terms of its *denoised* prediction $\hat{x}^{(1)}(x^{(t)})$. One can view this as the model being provided the noisy data $x^{(t)}$ for $t \in [0, 1)$ and predicting the *clean* data. The following objective will train the model to predict the marginal vector field

$$\mathbb{E}_{\mathcal{U}(t;0,1),p^{(1)}(x^{(1)}),p^{(0)}(x^{(0)})}\left[\|u(x^{(t)}|x^{(1)}) - \frac{\mathrm{Log}_{x^{(t)}}(\hat{x}^{(1)}(x^{(t)}))}{1-t}\|_2^2\right]. \tag{4.5}$$

We will use Eq. (4.5) as the form of the training objective henceforth. If the model is trained to low enough error across all t, then the learned vector field approximates the marginal vector field:

$$\underbrace{\hat{v}^{(t)}(x^{(t)}) = \frac{\text{Log}_{x^{(t)}}(\hat{x}^{(1)}(x^{(t)}))}{1-t}}_{\text{Learned}} \approx v^{(t)}. \tag{4.6}$$

Now with an approximation of the marginal vector field at hand, we can generate data by first drawing a sample from noise $x^{(0)} \sim p^{(0)}$ and simulating the ODE in Eq. (4.1) using the Euler method. First, we select a number of timesteps, e.g. 100. Then set the step size to $\Delta = 1/100$. The following update are performed starting at $t = 0.0$ until $t = 1.0$

$$x^{(t+\Delta)} = \text{Exp}_{x^{(t)}}\left(\Delta \cdot \text{Log}_{x^{(t)}}\left(\hat{v}^{(t)}(x^{(t)})\right)\right) \tag{4.7}$$

The final step is a sample from the learned data distribution: $x^{(1)} \sim p^{(1)}$.

Structure Noising

We describe the noising process for a biomolecule $B^{(1)}$ with a hybrid representation. The noising process is a function of the ground truth biomolecule $B^{(1)}$ and the noise time t. The pseudocode is provided in Algorithm 12.

Evaluation

We evaluate the generative model on the PoseBusters set [25]. We find that the generative model is able to sample structures with lower clash score and higher intra-ligand LDDT values indicating more native-like structures. We hypothesize that this is due to two reasons: 1) the noise breaks molecular symmetries between identical atoms in symmetric ligands, allowing the model to place symmetric atoms in different locations in predictions and 2) training across noise scales allows the model to learn more detailed features of the interactions between atoms (such as the ideal distance between atoms to avoid steric clashes) Figure 4.1.

Algorithm 12 NoiseBiomolecule

Require:
$B^{(1)}$ Ground truth biomolecule
t Noise time

$B^{(t)} \leftarrow []$
for $b^{(1)}$ in $B^{(1)}$ **do**
  if $b^{(1)}$ is frame **then**
    $(r^{(1)}, \tau^{(1)}) \leftarrow b_i^{(1)}$
    $r^{(0)} \sim \mathcal{U}(\mathrm{SO}(3))$
    $\tau^{(0)} \sim \mathcal{N}(\vec{0}, \mathrm{I}_3)$
    $r^{(t)} \leftarrow \mathrm{Exp}_{r^{(1)}}\left(t \cdot \mathrm{Log}_{r^{(1)}}(r^{(0)})\right)$
    $\tau^{(t)} \leftarrow t \cdot \tau^{(1)} + (1-t) \cdot \tau^{(0)}$
    $b^{(t)} \leftarrow (r^{(t)}, \tau^{(t)})$
  else if $b^{(1)}$ atomized or ligand atoms **then**
    $b^{(0)} \sim \mathcal{N}(b_i^{(1)}, \mathrm{I}_3)^{14}$
    $b^{(t)} \leftarrow t \cdot b_i^{(1)} + (1-t) \cdot b^{(0)}$
  end if
  $B^{(t)} \leftarrow \mathrm{Concat}(B^{(t)}, b^{(t)})$
end for
Return $B^{(t)}$

4.4 Discussion

The development of ModelHub shows the diversity of models that can be trained using a single software package. This flexibility is integral when experimenting with new model architectures and training procedures. We additionally are able to train models such as AF3 [2] with the same software package which were developed concurrently with our work and diverge quite substantially in their training procedures. We expect that this framework will be broadly useful for the community to develop and experiment with new models and to this end, we will open source the code under the MIT license.

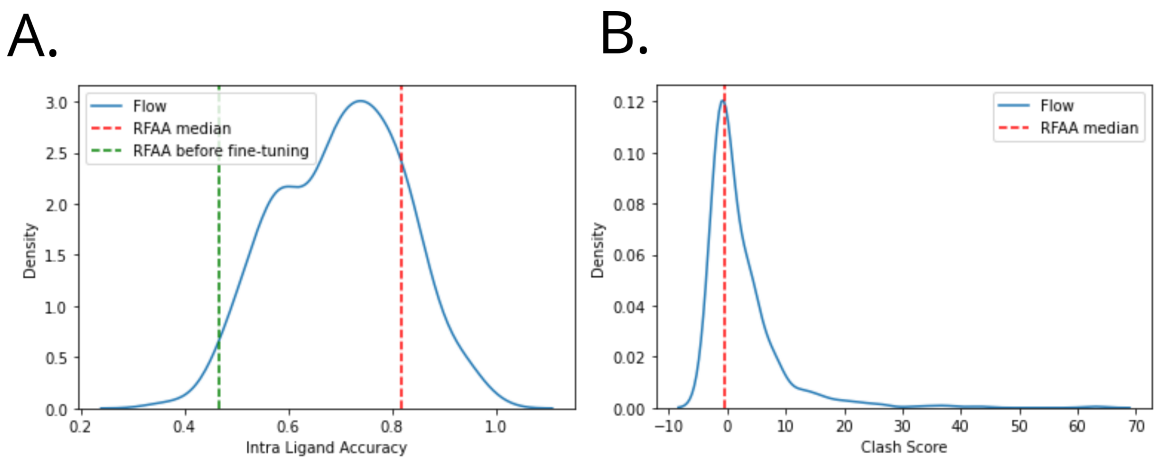


Figure 4.1: Evaluation of the generative model on the PoseBusters benchmark. **A**: The model is able to sample structures with higher intra-ligand LDDT values. Distribution shows the LDDT values of the predicted structures. Green dotted line represents the median prediction of RFAA [76] without specific fine-tuning to improve LDDT. The red dotted line represents the median prediction of RFAA with specific fine-tuning to improve LDDT. **B**: The model is able to sample structures with lower clash scores. Distribution shows the clash scores of the predicted structures. Dotted red line represents the median clash score of RFAA after fine-tuning with clash score as a loss.

Chapter 5

CONCLUSION AND FUTURE DIRECTIONS

This dissertation represents the first efforts to apply modern deep learning to biomolecular structure prediction and design beyond just proteins. At the outset of the work, it was unclear if the public data in the PDB [17] would be sufficient to learn about diverse biomolecular interactions such as those between small molecules and proteins. One insight that could be drawn from the results presented in this dissertation is that improved modeling techniques can improve model accuracy on smaller datasets. The specific techniques used will change but we believe that the approach of developing a single molecular modeling paradigm that is competent across multiple modalities will survive the test of time.

While these methods approach atomic accuracy, there is still considerable room for improvement in the atomic details of the predicted interactions. One avenue for improvement is to consider the interactions using a quantum mechanical framework which might be necessary to understand the detailed interactions between atoms in biomolecular systems. This could be either through neural network architectures that mimic quantum mechanical functions or models that predict quantum mechanical properties from structures. Quantum mechanical accuracy could be key to designing novel enzymes that have not been observed in nature.

Another avenue for improvement is including more data modalities into the model training regimen. Recent advances in Cryo-ET[19] allow for low-resolution *in situ* imaging of complex objects such as assymetric viruses or whole cells. This data poses many challenges: the resolution is often too low to resolve the atomic details of the interactions Modeling approaches that can fit atomic coordinates or distributions of atomic positions will be necessary to understand these systems. I anticipate that development of modeling techniques that can assess how higher order biomolecular function arises from atomic interaction will be a future

area of research.

This dissertation explores the initial applications of deep neural networks to prediction of biological structure. While this dissertation exclusively focuses on training neural networks, I believe that future paradigms in computational sciences could have a similar impact on our understanding of biology as the rise of deep learning. Future work will build on this foundation and move us towards a next generation of biological research fueled by computation.

BIBLIOGRAPHY

[1] UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Res.*, 51(D1):D523–D531, 2023.

[2] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

[3] Paul D. Adams, Pavel V. Afonine, Gábor Bunkóczi, Vincent B. Chen, Ian W. Davis, Nathaniel Echols, Jeffrey J. Headd, Li-Wei Hung, Gary J. Kapral, Ralf W. Grosse-Kunstleve, Airlie J. McCoy, Nigel W. Moriarty, Robert Oeffner, Randy J. Read, David C. Richardson, Jane S. Richardson, Thomas C. Terwilliger, and Peter H. Zwart. *PHENIX*: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallographica Section D*, 66(2):213–221, Feb 2010.

[4] N. Adir, S. Bar-Zvi, and D. Harris. The amazing phycobilisome. *Biochim. Biophys. Acta Bioenerg.*, 1861:148047, 2020.

[5] J. Adolf-Bryfogle, J. W. Labonte, J. C. Kraft, M. Shapavolov, S. Raemisch, T. Lütteke, F. DiMaio, C. D. Bahl, J. Pallesen, N. P. King, J. J. Gray, D. W. Kulp, and W. R. Schief. Growing glycans in rosetta: Accurate de novo glycan modeling, density fitting, and rational sequon design. *bioRxiv*, page 2021.09.27.462000, 2021.

[6] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[7] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel, J. W. Labonte, M. S.

Pacella, R. Bonneau, P. Bradley, R. L. Dunbrack Jr, R. Das, D. Baker, B. Kuhlman, T. Kortemme, and J. J. Gray. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13:3031–3048, 2017.

[8] Richard M Alvey, Avijit Biswas, Wendy M Schluchter, and Donald A Bryant. Attachment of noncognate chromophores to CpcA of synechocystis sp. PCC 6803 and synechococcus sp. PCC 7002 by heterologous expression in escherichia coli. *Biochemistry*, 50(22):4890–4902, June 2011.

[9] Ivan Anishchenko, Minkyung Baek, Hahnbeom Park, Naozumi Hiranuma, David E Kim, Justas Dauparas, Sanaa Mansoor, Ian R Humphreys, and David Baker. Protein tertiary structure prediction and refinement using deep learning and rosetta in CASP14. *Proteins*, 89(12):1722–1733, December 2021.

[10] Minkyung Baek, Ivan Anishchenko, Ian R Humphreys, Qian Cong, David Baker, and Frank DiMaio. Efficient and accurate prediction of protein structure using rosettafold2. *BioRxiv*, pages 2023–05, 2023.

[11] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carlos Adams, Craig R Glassman, Alexander DeGiovanni, Jose H Pereira, Andrew V Rodrigues, Amelie A van Dijk, Andrea C Ebrecht, D. J. Opperman, Tobias Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K Rathinaswamy, Ujwala Dalwadi, Christopher K Yip, John E Burke, K. Christopher Garcia, Nick V Grishin, Paul D Adams, Randy J Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[12] Minkyung Baek, Ryan McHugh, Ivan Anishchenko, David Baker, and Frank DiMaio. Accurate prediction of nucleic acid and protein-nucleic acid complexes using rosettafoldna. *bioRxiv*, page 2022.09.09.507333, 2022.

[13] H. Bagdonas, C. A. Fogarty, E. Fadda, and J. Agirre. The case for post-predictional modifications in the alphafold protein structure database. *Nature Structural and Molecular Biology*, 28:869–870, 2021.

[14] Christoph Bannwarth, Eike Caldeweyher, Sebastian Ehlert, Andreas Hansen, Philipp Pracht, Jakob Seibert, Sebastian Spicher, and Stefan Grimme. Extended tight-binding quantum chemistry methods. *WIREs Computational Molecular Science*, 11(2):e1493, 2021.

[15] S. F. H. Barnett, A. Hitchcock, A. K. Mandal, C. Vasilev, J. M. Yuen, J. Morby, A. A. Brindley, D. M. Niedzwiedzki, D. A. Bryant, A. J. Cadby, D. Holten, and C. N. Hunter. Repurposing a photosynthetic antenna protein as a super-resolution microscopy label. *Sci. Rep.*, 7:16807, 2017.

[16] Nathaniel R Bennett, Brian Coventry, Inna Goreshnik, Buwei Huang, Aza Allen, Dionne Vafeados, Ying Po Peng, Justas Dauparas, Minkyung Baek, Lance Stewart, Frank DiMaio, Steven De Munck, Savvas N Savvides, and David Baker. Improving de novo protein binder design with deep learning. *Nat. Commun.*, 14(1):2625, May 2023.

[17] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, T N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic Acids Res.*, 28(1):235–242, January 2000.

[18] Michele Bertoni, Florian Kiefer, Marco Biasini, Lukas Bordoli, and Torsten Schwede. Modeling protein quaternary structure of homo- and hetero-oligomers beyond binary interactions by homology. *Scientific Reports*, 7:10480, 2017.

[19] Tanmay A M Bharat and Sjors H W Scheres. Resolving macromolecular structures from electron cryo-tomography data using subtomogram averaging in relion. *Nature Protocols*, 11(11):2054–2065, 2016.

[20] Nick Bhattacharya, Neil Thomas, Roshan Rao, Justas Daupras, Peter K Koo, David Baker, Yun S Song, and Sergey Ovchinnikov. Single layers of attention suffice to predict protein contacts. October 2020.

[21] M. J. Bick, P. J. Greisen, K. J. Morey, M. S. Antunes, D. La, B. Sankaran, L. Reymond, K. Johnsson, J. I. Medford, and D. Baker. Computational design of environmental sensors for the potent opioid fentanyl. *eLife*, 6, 2017.

[22] S. Bienert, A. Waterhouse, T. A. P. de Beer, G. Tauriello, G. Studer, L. Bordoli, and T. Schwede. The swiss-model repository–new features and functionality. *Nucleic Acids Research*, 45:D313–D319, 2017.

[23] Markus Braun, Adrian Tripp, Morakot Chakatok, Sigrid Kaltenbrunner, Massimo Totaro, David Stoll, Aleksandar Bijelic, Wael Elaily, Shlomo Yakir Hoch, Matteo Aleotti, et al. Computational design of highly active de novo enzymes. *bioRxiv*, pages 2024–08, 2024.

[24] R Buller, S Lutz, RJ Kazlauskas, R Snajdrova, JC Moore, and UT Bornscheuer. From nature to industry: Harnessing enzymes for biocatalysis. *Science*, 382(6673):eadh8615, 2023.

[25] M. Buttenschoen, G. M. Morris, and C. M. Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *arXiv preprint*, 2023.

[26] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10:421, December 2009.

[27] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

[28] Longxing Cao, Brian Coventry, Inna Goreshnik, Buwei Huang, William Sheffler, Joon Sung Park, Kevin M Jude, Iva Marković, Rameshwar U Kadam, Koen H G Verschueren, Kenneth Verstraete, Scott Thomas Russell Walsh, Nathaniel Bennett, Ashish Phal, Aerin Yang, Lisa Kozodoy, Michelle DeWitt, Lora Picton, Lauren Miller, Eva-Maria Strauch, Nicholas D DeBouver, Allison Pires, Asim K Bera, Samer Halabiya,

Bradley Hammerson, Wei Yang, Steffen Bernard, Lance Stewart, Ian A Wilson, Hannele Ruohola-Baker, Joseph Schlessinger, Sangwon Lee, Savvas N Savvides, K Christopher Garcia, and David Baker. Design of protein-binding proteins from the target structure alone. *Nature*, 605(7910):551–560, May 2022.

[29] C. Chang, C. Lee Luo, and Y. Gao. In crystallo observation of three metal ion promoted dna polymerase misincorporation. *Nature Communications*, 13:2346, 2022.

[30] I-Min A Chen, Victor M Markowitz, Ken Chu, Krishna Palaniappan, Ernest Szeto, Manoj Pillay, Anna Ratner, Jinghua Huang, Evan Andersen, Marcel Huntemann, Neha Varghese, Michalis Hadjithomas, Kristin Tennessen, Torben Nielsen, Natalia N Ivanova, and Nikos C Kyrpides. IMG/M: integrated genome and metagenome comparative data analysis system. *Nucleic Acids Res.*, 45(D1):D507–D516, January 2017.

[31] Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

[32] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[33] Alexander E Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024.

[34] G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

[35] Bobo Dang, Marco Mravic, Hailin Hu, Nathan Schmidt, Bruk Mensa, and William F DeGrado. SNAC-tag for sequence-specific chemical protein cleavage. *Nature Methods*, 16(4):319–322, April 2019.

[36] R. Das and D. Baker. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.*, 77:363–382, 2008.

[37] J Dauparas, I Anishchenko, N Bennett, H Bai, R J Ragotte, L F Milles, B I M Wicky, A Courbet, R J de Haas, N Bethel, P J Y Leung, T F Huddy, S Pellock, D Tischer, F Chan, B Koepnick, H Nguyen, A Kang, B Sankaran, A K Bera, N P King, and D Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

[38] Justas Dauparas, Gyu Rie Lee, Robert Pecoraro, Linna An, Ivan Anishchenko, Cameron Glasscock, and David Baker. Atomic context-conditioned protein sequence design using ligandmpnn. *Nature Methods*, pages 1–7, 2025.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[40] Digitalis Investigation Group. The effect of digoxin on mortality and morbidity in patients with heart failure. *N. Engl. J. Med.*, 336:525–533, 1997.

[41] A. M. Díaz-Rovira, H. Martín, T. Beuming, L. Díaz, V. Guallar, and S. S. Ray. Are deep learning structural models sufficiently accurate for virtual screening? application of docking algorithms to alphafold2 predicted structures. *Journal of Chemical Information and Modeling*, 63:1668–1674, 2023.

[42] J. Eberhardt, D. Santos-Martins, A. F. Tillack, and S. Forli. Autodock vina 1.2.0: New docking methods, expanded force field, and python bindings. *Journal of Chemical Information and Modeling*, 61:3891–3898, 2021.

[43] Sean R Eddy. Accelerated profile HMM searches. *PLoS Comput. Biol.*, 7(10):e1002195, October 2011.

[44] Paul Emsley and Kevin Cowtan. *Coot*: model-building tools for molecular graphics. *Acta Crystallographica Section D*, 60(12 Part 1):2126–2132, Dec 2004.

[45] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[46] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Timothy Green, Augustin Žídek, Russell Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Mirko Zielinski, Andrej Bridgland, Alexander Potapenko, Charlie Cowie, Kathryn Tunyasuvunakool, Ruchi Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with alphafold-multimer. *bioRxiv*, page 2021.10.04.463034, 2022.

[47] R. J. Flanagan and A. L. Jones. Fab antibody fragments: some applications in clinical toxicology. *Drug Saf.*, 27:1115–1133, 2004.

[48] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis, and P. S. Shenkin. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47:1739–1749, 2004.

[49] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski,

R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. Gaussian˜16 Revision C.01, 2016. Gaussian Inc. Wallingford CT.

[50] Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. SE(3)-Transformers: 3D Roto-Translation equivariant attention networks. June 2020.

[51] Elisabeth Gasteiger, Christine Hoogland, Alexandre Gattiker, S'everine Duvaud, Marc R. Wilkins, Ron D. Appel, and Amos Bairoch. *Protein Identification and Analysis Tools on the ExPASy Server*, pages 571–607. Humana Press, Totowa, NJ, 2005.

[52] A. Gorelik, K. Illes, K. H. Bui, and B. Nagar. Structures of the mannose-6-phosphate pathway enzyme, glcnac-1-phosphotransferase. *Proceedings of the National Academy of Sciences of the United States of America*, 119:e2203518119, 2022.

[53] Stefan Grimme, Stephan Ehrlich, and Lars Goerigk. Effect of the damping function in dispersion corrected density functional theory. *Journal of Computational Chemistry*, 32(7):1456–1465, 2011.

[54] C. R. Groom, I. J. Bruno, M. P. Lightfoot, and S. C. Ward. The cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 72:171–179, 2016.

[55] Aurelien Grosdidier, Vincent Zoete, and Olivier Michielin. Swissdock, a protein-small molecule docking web service based on eadock dss. *Nucleic acids research*, 39(suppl_2):W270–W277, 2011.

[56] J. Haas, A. Barbato, D. Behringer, G. Studer, S. Roth, M. Bertoni, K. Mostaguir, R. Gumienny, and T. Schwede. Continuous automated model evaluation (cameo) complementing the critical assessment of structure prediction in casp12. *Proteins*, 86 Suppl 1:387–398, 2018.

[57] J. Haas, R. Gumienny, A. Barbato, F. Ackermann, G. Tauriello, M. Bertoni, G. Studer, A. Smolinski, and T. Schwede. Introducing "best single template" models as reference

baseline for the continuous automated model evaluation (cameo). *Proteins*, 87:1378–1387, 2019.

[58] J. Haas, S. Roth, K. Arnold, F. Kiefer, T. Schmidt, L. Bordoli, and T. Schwede. The protein model portal–a comprehensive resource for protein structure and model information. *Database*, page bat031, 2013.

[59] M. L. Hekkelman, I. de Vries, R. P. Joosten, and A. Perrakis. Alphafill: enriching alphafold models with ligands and cofactors. *Nature Methods*, 20:205–213, 2023.

[60] A. Hitchcock, C. N. Hunter, R. Sobotka, J. Komenda, M. Dann, and D. Leister. Redesigning the photosynthetic light reactions to enhance photosynthesis - the photoredesign consortium. *Plant J.*, 109:23–34, 2022.

[61] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[62] M. Holcomb, Y.-T. Chang, D. S. Goodsell, and S. Forli. Evaluation of alphafold2 structures as docking targets. *Protein Science*, 32:e4530, 2023.

[63] R. V. Honorato, J. Roel-Touris, and A. M. J. J. Bonvin. Martini-based protein-dna coarse-grained haddocking. *Frontiers in Molecular Biosciences*, 6:102, 2019.

[64] Euan J Hossack, Florence J Hardy, and Anthony P Green. Building enzymes through design and evolution. *ACS Catalysis*, 13(19):12436–12444, 2023.

[65] KN Houk and Fang Liu. Holy grails for computational organic chemistry and biochemistry. *Accounts of chemical research*, 50(3):539–543, 2017.

[66] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8946–8970. PMLR, 2022.

[67] X. Huang and J. T. Groves. Oxygen activation and radical transformations in heme proteins and metalloporphyrins. *Chem. Rev.*, 118:2491–2553, 2018.

[68] Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein backbone generation. *arXiv preprint arXiv:2405.20313*, 2024.

[69] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[70] J. Ingraham, M. Baranov, Z. Costello, V. Frappier, A. Ismail, S. Tie, W. Wang, V. Xue, F. Obermeyer, A. Beam, and G. Grigoryan. Illuminating protein space with a programmable generative model. *bioRxiv*, 2022. Preprint, doi: 10.1101/2022.12.01.518682.

[71] S. Jo, H. S. Lee, J. Skolnick, and W. Im. Restricted n-glycan conformational space in the pdb and its implication in glycan structure modeling. *PLoS Computational Biology*, 9(4):e1002946, 2013.

[72] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, et al. Alphafold 2. *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction*, 2020.

[73] Wolfgang Kabsch. *XDS*. *Acta Crystallographica Section D*, 66(2):125–132, Feb 2010.

[74] I. Kalvet, M. Ortmayer, J. Zhao, R. Crawshaw, N. M. Ennist, C. Levy, A. Roy, A. P. Green, and D. Baker. Design of heme enzymes with a tunable substrate binding pocket adjacent to an open metal coordination site. *J. Am. Chem. Soc.*, 145:14307–14315, 2023.

[75] E. Konia, K. Chatzicharalampous, A. Drakonaki, C. Muenke, U. Ermler, G. Tsiotis, and I. V. Pavlidis. Rational engineering of luminiphilus syltensis (r)-selective amine transaminase for the acceptance of bulky substrates. *Chem. Commun.*, 57:12948–12951, 2021.

[76] Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

[77] Anna Lauko, Samuel J Pellock, Ivan Anischanka, Kiera H Sumida, David Juergens, Woody Ahern, Alex Shida, Andrew Hunt, Indrek Kalvet, Christoffer Norn, et al. Computational design of serine hydrolases. *bioRxiv*, 2024.

[78] K. Le, M. J. Soth, J. B. Cross, G. Liu, W. J. Ray, J. Ma, S. G. Goodwani, P. J. Acton, V. Buggia-Prevot, O. Akkermans, J. Barker, M. L. Conner, Y. Jiang, Z. Liu, P. McEwan, J. Warner-Schmidt, A. Xu, M. Zebisch, C. J. Heijnen, B. Abrahams, and P. Jones. Discovery of iacs-52825, a potent and selective dlk inhibitor for treatment of chemotherapy-induced peripheral neuropathy. *J. Med. Chem.*, 66:9954–9971, 2023.

[79] Gyu Rie Lee, Samuel J. Pellock, Christoffer Norn, Doug Tischer, Justas Dauparas, Ivan Anischenko, Jaron A. M. Mercer, Alex Kang, Asim Bera, Hannah Nguyen, Inna Goreshnik, Dionne Vafeados, Nicole Roullier, Hannah L. Han, Brian Coventry, Hugh K. Haddox, David R. Liu, Andy Hsien-Wei Yeh, and David Baker. Small-molecule binding and sensing with a designed protein family. *bioRxiv*, 2023.

[80] J. M. Lee, H. M. Hammarén, M. M. Savitski, and S. H. Baek. Control of protein stability by post-translational modifications. *Nature Communications*, 14:201, 2023.

[81] Z. Liao, R. You, X. Huang, and X. Yao. Deepdock: Enhancing ligand-protein interaction prediction by a combination of ligand and structure information. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1852–1857, 2019.

[82] Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

[83] Zeming Lin, Hannes Akin, Roshan Rao, Brian Hie, Zongyi Zhu, William Lu, Nikita Smetanin, Rianne Verkuil, Ori Kabeli, Yannan Shmueli, Ana dos Santos Costa, Mohammad Fazel-Zarandi, Tom Sercu, Serkan Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379:1123–1130, 2023.

[84] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[85] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[86] Sarah L Lovelock, Rebecca Crawshaw, Sophie Basler, Colin Levy, David Baker, Donald Hilvert, and Anthony P Green. The road to fully programmable protein catalysis. *Nature*, 606(7912):49–58, 2022.

[87] W. Lu, Q. Wu, J. Zhang, J. Rao, C. Li, S. Zheng, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *bioRxiv*, pages 7236–7249, 2022.

[88] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[89] J. A. Mancini, M. Sheehan, G. Kodali, B. Y. Chow, D. A. Bryant, P. L. Dutton, and C. C. Moser. De novo synthetic biliprotein design, assembly and excitation energy transfer. *J. R. Soc. Interface*, 15, 2018.

[90] A. Marx and N. Adir. Allophycocyanin and phycocyanin crystal structures reveal facets of phycobilisome assembly. *Biochim. Biophys. Acta*, 1827:311–318, 2013.

[91] Airlie J. McCoy, Ralf W. Grosse-Kunstleve, Paul D. Adams, Martyn D. Winn, Laurent C. Storoni, and Randy J. Read. *Phaser* crystallographic software. *Journal of Applied Crystallography*, 40(4):658–674, Aug 2007.

[92] B. Ni, D. L. Kaplan, and M. J. Buehler. Generative design of de novo proteins based on secondary structure constraints using an attention-based diffusion model. *Chem*, 9:1828–1849, 2023.

[93] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *J. Cheminform.*, 3:33, October 2011.

[94] H. Park, G. Zhou, M. Baek, D. Baker, and F. DiMaio. Force field optimization guided by small molecule crystal lattice data enables consistent sub-angstrom protein-ligand docking. *Journal of Chemical Theory and Computation*, 17:2000–2010, 2021.

[95] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, High-Performance deep learning library. December 2019.

[96] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[97] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[98] N. F. Polizzi and W. F. DeGrado. A defined structural unit enables de novo design of small-molecule-binding proteins. *Science*, 369:1227–1233, 2020.

[99] T. L. Poulos. Heme enzyme structure and function. *Chem. Rev.*, 114:3919–3962, 2014.

[100] Z. Qiao, W. Nie, A. Vahdat, T. F. Miller III, and A. Anandkumar. State-specific

protein-ligand complex structure prediction with a multi-scale deep generative model. *arXiv preprint*, 2022.

[101] L. L. Rade, W. C. Generoso, S. Das, A. S. Souza, R. L. Silveira, M. C. Avila, P. S. Vieira, R. Y. Miyamoto, A. B. B. Lima, J. A. Aricetti, R. R. de Melo, N. Milan, G. F. Persinoti, A. M. F. L. J. Bonomi, M. T. Murakami, T. M. Makris, and L. M. Zanphorlin. Dimer-assisted mechanism of (un)saturated fatty acid decarboxylation for alkene production. *Proc. Natl. Acad. Sci. U. S. A.*, 120:e2221483120, 2023.

[102] S. Ramazi and J. Zahiri. Posttranslational modifications in proteins: resources, tools and prediction methods. *Database*, 2021, 2021.

[103] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA transformer. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR, 2021.

[104] A K Rappe, C J Casewit, K S Colwell, W A Goddard, Iii, and W M Skiff. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *J. Am. Chem. Soc.*, 114(25):10024–10035, December 1992.

[105] K. Raja Reddy, M. Totrov, O. Lomovskaya, D. C. Griffith, Z. Tarazi, M. C. Clifton, and S. J. Hecker. Broad-spectrum cyclic boronate beta-lactamase inhibitors featuring an intramolecular prodrug for oral bioavailability. *Bioorg. Med. Chem.*, 62:116722, 2022.

[106] C. Reily, T. J. Stewart, M. B. Renfrow, and J. Novak. Glycosylation in health and disease. *Nature Reviews Nephrology*, 15:346–366, 2019.

[107] Julia C Reisenbauer, Kathleen M Sicinski, and Frances H Arnold. Catalyzing the future: recent advances in chemical synthesis using enzymes. *Current opinion in chemical biology*, 83:102536, 2024.

[108] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. HHblits:

lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods*, 9(2):173–175, December 2011.

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

[110] A. Schenkmayerova, M. Toul, D. Pluskal, R. Baatallah, G. Gagnot, G. P. Pinto, V. T. Santana, M. Stuchla, P. Neugebauer, P. Chaiyen, J. Damborsky, D. Bednar, Y. L. Janin, Z. Prokop, and M. Marek. Catalytic mechanism for renilla-type luciferases. *Nature Catalysis*, 6:23–38, 2023.

[111] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[112] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[113] M. Sono, J. H. Dawson, and L. P. Hager. The generation of a hyperporphyrin spectrum upon thiol binding to ferric chloroperoxidase. further evidence of endogenous thiolate ligation to the ferric enzyme. *J. Biol. Chem.*, 259:13209–13216, 1984.

[114] M Sono, J H Dawson, and L P Hager. The generation of a hyperporphyrin spectrum upon thiol binding to ferric chloroperoxidase. further evidence of endogenous thiolate ligation to the ferric enzyme. *Journal of Biological Chemistry*, 259(21):13209–13216, 1984.

[115] Martin Steinegger, Markus Meier, Milot Mirdita, Harald Vöhringer, Stephan J Haunsberger, and Johannes Söding. HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics*, 20(1):473, September 2019.

[116] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, 35(11):1026–1028, October 2017.

[117] Gabriel Studer, Christoph Rempfer, Andrew M. Waterhouse, Rafal Gumienny, Johannes Haas, and Torsten Schwede. Qmeandisco–distance constraints applied on model quality estimation. *Bioinformatics*, 36:1765–1771, 2020.

[118] H. Stärk, O.-E. Ganea, L. Pattanaik, R. Barzilay, T. Jaakkola, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato. Equibind: Geometric deep learning for drug binding structure prediction. In *Proceedings of the 39th International Conference on Machine Learning*, pages 20503–20521, 2022.

[119] S. Tang, Y. Lu, W. M. Skinner, M. Sanyal, P. V. Lishko, M. Ikawa, and P. S. Kim. Human sperm tmem95 binds eggs and facilitates membrane fusion. *Proceedings of the National Academy of Sciences of the United States of America*, 119:e2207805119, 2022.

[120] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. February 2018.

[121] Matthew Z Tien, Austin G Meyer, Dariya K Sydykova, Stephanie J Spielman, and Claus O Wilke. Maximum allowed solvent accessibilites of residues in proteins. *PloS one*, 8(11):e80635, 2013.

[122] C. E. Tinberg, S. D. Khare, J. Dou, L. Doyle, J. W. Nelson, A. Schena, W. Jankowski, C. G. Kalodimos, K. Johnsson, B. L. Stoddard, and D. Baker. Computational design of ligand-binding proteins with high affinity and selectivity. *Nature*, 501:212–216, 2013.

[123] Susana Vázquez Torres, Philip J Y Leung, Isaac D Lutz, Preetham Venkatesh, Joseph L Watson, Fabian Hink, Huu-Hien Huynh, Andy Hsien-Wei Yeh, David Juergens,

Nathaniel R Bennett, Andrew N Hoofnagle, Eric Huang, Michael J MacCoss, Marc Expòsit, Gyu Rie Lee, Paul M Levine, Xinting Li, Mila Lamb, Elif Nihal Korkmaz, Jeff Nivala, Lance Stewart, Joseph M Rogers, and David Baker. De novo design of high-affinity protein binders to bioactive helical peptides. December 2022.

[124] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[125] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, and S. Velankar. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50:D439–D444, 2022.

[126] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[127] Jue Wang, Sidney Lisanza, David Juergens, Doug Tischer, Joseph L Watson, Karla M Castro, Robert Ragotte, Amijai Saragovi, Lukas F Milles, Minkyung Baek, Ivan Anishchenko, Wei Yang, Derrick R Hicks, Marc Expòsit, Thomas Schlichthaerle, Jung-Ho Chun, Justas Dauparas, Nathaniel Bennett, Basile I M Wicky, Andrew Muenks, Frank DiMaio, Bruno Correia, Sergey Ovchinnikov, and David Baker. Scaffolding protein functional sites using deep learning. *Science*, 377(6604):387–394, July 2022.

[128] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The PDBbind database: Collection of binding affinities for Protein-Ligand complexes with known Three-Dimensional structures. *J. Med. Chem.*, 47(12):2977–2980, June 2004.

[129] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Computational Biology*, 13(1):e1005324, 2017.

[130] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[131] Andrew Waterhouse, Michele Bertoni, Stefan Bienert, Gabriel Studer, Giuseppe Tauriello, Rafal Gumienny, Florian T. Heer, Tjaart A. P. de Beer, Christoph Rempfer, Lukas Bordoli, Roberta Lepore, and Torsten Schwede. Swiss-model: homology modelling of protein structures and complexes. *Nucleic Acids Research*, 46:W296–W303, 2018.

[132] Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv*, 2022.

[133] B I M Wicky, L F Milles, A Courbet, R J Ragotte, J Dauparas, E Kinfu, S Tipps, R D Kibler, M Baek, F DiMaio, X Li, L Carter, A Kang, H Nguyen, A K Bera, and D Baker. Hallucinating symmetric protein assemblies. *Science*, 378(6615):56–61, October 2022.

[134] Christopher J. Williams, Jeffrey J. Headd, Nigel W. Moriarty, Michael G. Prisant, Lizbeth L. Videau, Lindsay N. Deis, Vishal Verma, Daniel A. Keedy, Bradley J. Hintze, Vincent B. Chen, Swati Jain, Steven M. Lewis, W. Bryan Arendall III, Jack Snoeyink,

Paul D. Adams, Simon C. Lovell, Jane S. Richardson, and David C. Richardson. Molprobity: More and better reference data for improved all-atom structure validation. *Protein Science*, 27(1):293–315, 2018.

[135] 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.

[136] R. J. Woods. Predicting the structures of glycans, glycoproteins, and their complexes. *Chemical Reviews*, 118(17):8005–8024, 2018.

[137] K.-L. Wu, J. A. Moore, M. D. Miller, Y. Chen, C. Lee, W. Xu, Z. Peng, Q. Duan, G. N. Phillips Jr, R. A. Uribe, and H. Xiao. Expanding the eukaryotic genetic code with a biosynthesized 21st amino acid. *Protein Sci.*, 31:e4443, 2022.

[138] L. Wu, B. L. Trippe, C. A. Naesseth, D. M. Blei, and J. P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *arXiv preprint*, 2023.

[139] Ruiqiang Wu, Fei Ding, Ruibo Wang, Rui Shen, Xiang Zhang, Song Luo, Chengxin Su, Zhiye Wu, Qiangfeng Cliff Xie, Bonnie Berger, Jian Peng, and Jinbo Xu. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, page 2022.07.21.500999, 2022.

[140] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, 2020.

[141] Jianyi Yang, Ambrish Roy, and Yang Zhang. BioLiP: a semi-manually curated database for biologically relevant ligand-protein interactions. *Nucleic Acids Res.*, 41(Database issue):D1096–103, January 2013.

[142] Andy Hsien-Wei Yeh, Christoffer Norn, Yakov Kipnis, Doug Tischer, Samuel J Pellock, Declan Evans, Pengchen Ma, Gyu Rie Lee, Jason Z Zhang, Ivan Anishchenko, et al. De novo design of luciferases using deep learning. *Nature*, 614(7949):774–780, 2023.

[143] Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

[144] Jason Yim, Andrew Campbell, Emile Mathieu, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Frank Noé, et al. Improved motif-scaffolding with se (3) flow matching. *ArXiv*, pages arXiv–2401, 2024.

[145] Y. Yuan, G. Jia, C. Wu, W. Wang, L. Cheng, Q. Li, Z. Li, K. Luo, S. Yang, W. Yan, Z. Su, and Z. Shao. Structures of signaling complexes of lipid receptors s1pr1 and s1pr5 reveal mechanisms of activation and drug recognition. *Cell Res.*, 31:1263–1274, 2021.

[146] Alexandre Zanghellini, Lin Jiang, Andrew M Wollacott, Gong Cheng, Jens Meiler, Eric A Althoff, Daniela Röthlisberger, and David Baker. New algorithms and an in silico benchmark for computational enzyme design. *Protein Science*, 15(12):2785–2794, 2006.

[147] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins*, 57(4):702–710, December 2004.

[148] C. Zhao, A. Höppner, Q.-Z. Xu, W. Gärtner, H. Scheer, M. Zhou, and K.-H. Zhao. Structures and enzymatic mechanisms of phycobiliprotein lyases cpce/f and pece/f. *Proc. Natl. Acad. Sci. U. S. A.*, 114:13170–13175, 2017.

[149] G. Zhou, Z. Gao, Q. Ding, H. Zheng, H. Xu, Z. Wei, L. Zhang, and G. Ke. Uni-mol: A universal 3d molecular representation learning framework. *ChemRxiv*, 2022.