Deep Learning Methods and Datasets for All-atom Protein Structure Prediction

by

Jonathan Edward King

Bachelor of Science in Bioengineering, University of California, Berkeley, 2017

Bachelor of Arts in Computer Science, University of California, Berkeley, 2017

Submitted to the Graduate Faculty of the

School of Medicine

in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This dissertation was presented

by

Jonathan Edward King

It was defended on

10-13-2023

and approved by

Olexandr Isayev, Associate Professor, Department of Chemistry

Andrew P. VanDemark, Associate Professor, Department of Biological Sciences

Jacob Durrant, Associate Professor, Department of Biological Sciences

Dissertation Director:
David R. Koes, Associate Professor, Department of Computational and Systems Biology

Deep Learning Methods and Datasets for All-atom Protein Structure Prediction

Jonathan Edward King, PhD

University of Pittsburgh, 2023

Over the past decade, significant progress has been made in protein structure prediction, largely thanks to tools like AlphaFold2. This paper tackles several lingering challenges in this field. First, we introduce SidechainNet, a dataset and toolkit designed to streamline the handling of protein sequence and structure data for machine learning and increase its accessibility. This initiative addresses the prevalent issue of effectively collecting and organizing data, especially in the realm of protein science, where data quality and availability may vary. Furthermore, leading methods have been observed to fall short in real-world tasks like molecular docking. To enhance the physical realism of predictions made by deep learning models, we implement potential energy as a loss function through OpenMM-Loss. This technique reduces potential energy and clashes in predicted structures, potentially rendering these predictions more viable for various applications. We also scrutinize AlphaFold2 with the aim of refining its sidechain modeling—a crucial aspect of drug discovery. Although we don't pinpoint a significantly more accurate model, our analysis reveals comparable performance between ResNet and Transformer models in sidechain prediction tasks. In light of these results, we recommend that future efforts concentrate on more holistic sidechain modeling efforts. Finally, we discuss potential future developments and extensions of our methods.

Table of Contents

List of Tables

List of Figures

Preface

I want to start by acknowledging all of the individuals who have given me tremendous support and encouragement.

To my partner, Katya, who has been by my side (though for several long years, at a distance) at every step of my academic and personal life: I could not have finished this work without your encouragement and shared enthusiasm. I am happy we found our intellectual passions together and even happier that those passions finally have their corresponding terminal degrees.

To my parents, David and Kathleen, to my brother, Andrew, and to my extended family members who have supported me throughout my life: you have taught me how to persevere, ask questions, and be kind above all. Thank you for your years of support, willingness to listen, compassion, and encouragement to study and focus when I felt like doing otherwise.

I also express my deepest gratitude to my advisor, David Koes, for his constant guidance and willingness to accept me into his research group. I've learned much from our work together, especially how to stay positive (and never give up on debugging). Thank you to my committee members who have provided consistent and thoughtful feedback.

Thank you also to my friends, colleagues, and lab mates (too many to mention individually, though Paul Francoeur, my six-year-cubicle-mate and friend, may get an extra nod), who made it all worth it. I'm not sure I knew just how long this would take, but I'm glad I did it!

Pittsburgh, September 2023 J. E. King

1.0 Introduction

The field of computational biology has seen remarkable advancements in recent years, owing much of its progress in some areas to the application of machine learning techniques. Among the various areas within computational biology, one of the most transformative breakthroughs has been the direct prediction of complete protein structures. Such advances can potentially revolutionize structure-based drug discovery, where knowledge of protein structures is pivotal for designing targeted medicines.

Until recently, protein structure prediction methods and their assessments have focused primarily on modeling the protein backbone. However, we hypothesize that predictive accuracy and quality can be significantly improved by incorporating the modeling of amino acid sidechain conformations alongside the protein backbone. This improvement is driven by the intricate relationship between amino acid orientation and the overall protein structure. Furthermore, this additional information can facilitate the direct computation of the energy of predicted structures, which we will show can be used as a valuable loss function for training deep learning models. In essence, our research seeks to elevate the state-of-the-art predictive performance within the all-atom protein structure prediction domain. We endeavor to reach this goal through three key aims:

1. Create and distribute a machine learning-oriented dataset for all-atom protein structures, coupled with tools to streamline machine learning model development;

2. Incorporate biophysical properties and force field-based training procedures to enhance the physical qualities of protein structure predictions; and

3. Integrate modeling methods specific to all-atom protein representations to enhance the predictive power of state-of-the-art predictive models, using established models like AlphaFold2 as benchmarks.

1.1 Related Work

1.1.1 The Rise of Deep Learning

The recent explosion in the application of deep learning techniques, initially popularized in natural language processing (NLP) and computer vision (CV), has left many researchers contemplating how to harness these advancements for diverse scientific domains, including structural biology. Several factors have contributed to the rapid adoption of deep learning in NLP and CV, including the availability of vast textual and image datasets on the internet, the recognition of commercial applications in various sectors, and the expansive landscape of publicly accessible competitions driving technological innovation.

However, this adoption has not been as straightforward in the natural sciences. Scientific data, particularly biological, chemical, and medical data, often lacks the abundance and accessibility of text and image data. Additionally, the inherent costs and complexities associated with laboratory-based research can divert early research efforts toward fields with lower initial investment costs, despite potential interest in other scientific domains. Moreover, the limited availability of benchmark datasets and competitions in the natural sciences has hindered the integration of machine learning techniques in these domains.

Despite these challenges, the recent convergence of data, computational resources, and analytical methods in biology signifies a transformative moment. This pivotal shift has enabled scientists to explore applications of machine learning methods beyond NLP and CV and to begin harnessing these powerful techniques for groundbreaking discoveries in the biological sciences. Building on this momentum, this dissertation poses and seeks to address the question: "How can we leverage deep learning to expand our understanding of protein biology, improve drug discovery, and make impactful scientific inferences?"

1.1.2 Protein Structure Prediction

Accurate protein structure prediction remains a profound challenge in molecular biology. Since the pioneering work of Kendrew and Perutz in determining the structure of sperm whale myoglobin through X-ray crystallography in the late 1950s[1], scientists have sought to unveil the 3D structures of numerous proteins to comprehend essential molecular processes. Proteins, composed of amino acids, may naturally fold into unique, globular structures vital for various cellular functions. Notably, knowledge of precise amino acid sidechain orientations is critical in structure-based drug discovery, which aims to develop medications by rational examination of protein-drug interactions.

However, acquiring experimental protein structure data is costly and labor-intensive, often with low success rates[2]. Consequently, the scientific community has dedicated substantial efforts to developing methods for predicting protein structures from primary amino acid sequences, given the vast availability of sequence data. This dichotomy is evident when comparing the abundance of protein sequences (over 250 million) in databases like UniProtKB[3] to the relatively sparse protein structures (about 200 thousand) in the Protein Data Bank[4] (PDB).

Protein structure prediction methods can be broadly categorized into template-based and template-free modeling. The former relies on known protein structures with similar sequences as references, while the latter, also known as *ab initio* modeling, generates structures without reference templates. Within these categories are more specific methodologies, including conformational sampling, fragment assembly, energy minimization, refinement, local (secondary structure, torsion angle) structure prediction, and, lastly, contact prediction. While these approaches have been the subject of rigorous evaluation through initiatives like the Critical Assessment of protein Structure Prediction (CASP) competition, challenges persist. For instance, the prediction of protein backbone conformations and sidechain orientations (which, together, make up all-atom protein structures) remains computationally demanding due to

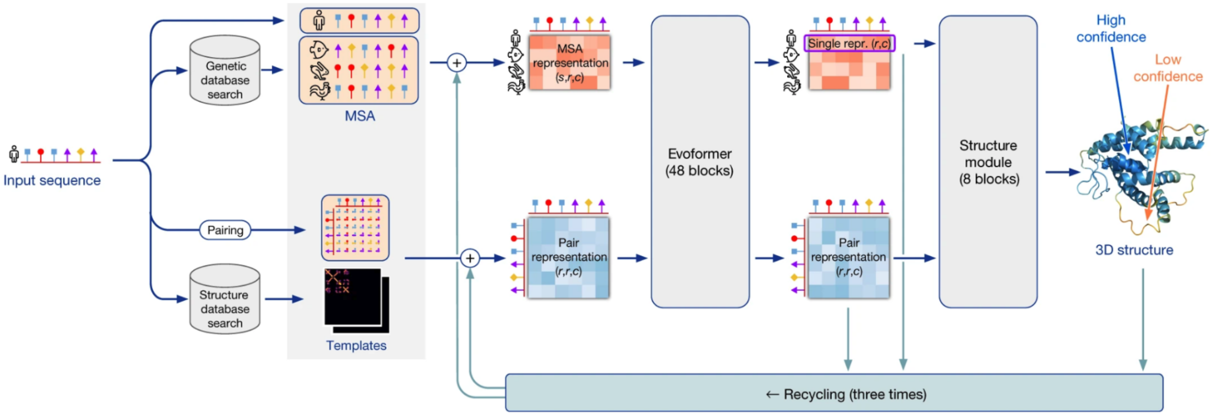

Input sequence
Genetic database search
Pairing
Structure database search
MSA
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Figure 1.1: The architecture of AlphaFold2[6], reproduced with permission.

its NP-hard nature[5].

The advent of AlphaFold[7] and AlphaFold2[6] (AF2) (Figure 1.1) in 2018 and 2020, respectively, signaled a pivotal shift in addressing these challenges. AF2, with its innovative attention-based architecture, exhibited unprecedented predictive performance in CASP competitions. However, AF2 is not without its shortcomings. It is heavily reliant on high-quality sequence data, and its predictions often necessitate energetic minimization due to their energetically unfavorable nature. Additionally, AF2 does not consider ligand information, rendering it incapable of adjusting predictions based on the presence or absence of a bound ligand. This oversight limits its utility in tasks like molecular docking or certain drug discovery processes. Nevertheless, while criticisms and limitations of AF2 persist, its breakthroughs have undeniably reinvigorated the pursuit of accurate protein structure prediction and served as a source of inspiration for the broader research community.

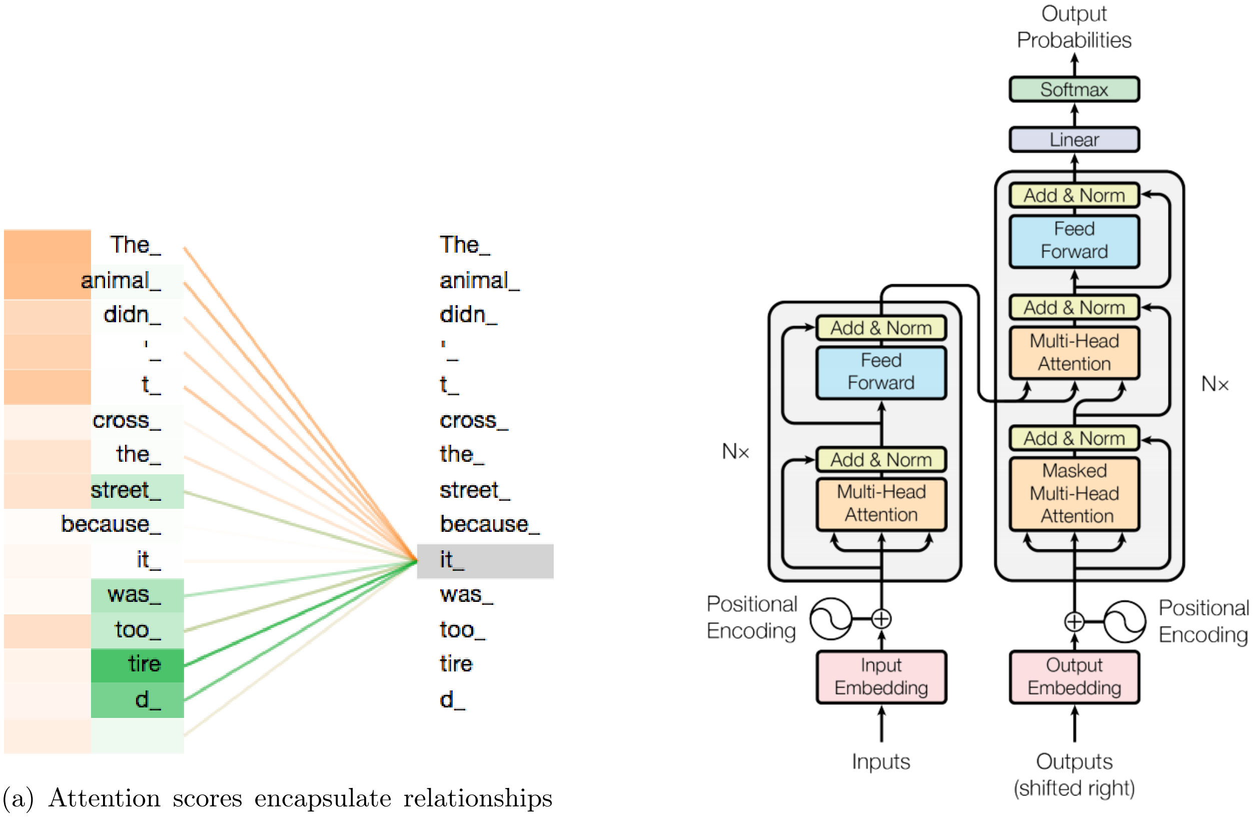
1.1.3 Deep Learning's Contribution to the Field

Deep learning, a subfield of machine learning, leverages neural networks with multiple layers to approximate complex functions. The success of deep learning is underpinned by the Universal Approximation Theorem, which states that a neural network with a single hidden layer, finite number of neurons, and non-linear linear activation function can approximate any continuous function. In practice, deep learning models improve their parameters via backpropagation and are trained on large sets of input-output pairs. This methodology has yielded exceptional results across a spectrum of applications, from image recognition[8] and autonomous navigation[9] to language translation[10].

The inspiration for this dissertation extends from Neural Machine Translation (NMT), a branch of NLP that employs neural networks to translate between languages. Although we do not explicitly employ our original idea in a separate aim, our early work investigated how to frame the problem of protein structure prediction as one of language translation: translating from the "language of amino acids" that describe a protein's sequence to the "language of torsional angles" that describe a protein's structure.

NMT introduced two dominant technologies: Recurrent Neural Networks[11–13] (RNNs) and Transformer[14–16] networks. RNNs process data sequentially, utilizing internal states to accumulate information and generate output sequences. Transformers, in contrast, operate in parallel, employing attention mechanisms to process entire input sequences simultaneously (Figures 1.2b and 1.2a). To create a new representation for each sequence element, it first compares, or scores, it with respect to every other sequence element (Figure 1.2a). The sequence elements are then scaled by their scores and added, creating a new representation of each sequence element as a weighted average over all other sequence elements. What makes this weighted average unique, however, is that scores and weights are predicted by the model. This causes the model to "attend" to only what it considers to be the relevant portions of the sequence for each element.

Transformers have gained preeminence in sequence-based deep learning due to their superior capacity for modeling long-range dependencies and lower computational demands. The parallel processing nature of Transformers reduces training times substantially, offering a significant advantage over RNNs. Researchers in protein structure prediction have closely monitored developments in NLP and applied these transformative technologies when applica-



(a) Attention scores encapsulate relationships between words. Figure by Jay Alammar[17], reproduced with permission..

(b) The Transformer architecture. Figure from Vaswani et al.[14], reproduced with permission.

Figure 1.2: Attention and Transformer Architectures.

ble. Notably, the success of AlphaFold2 and the adoption of Transformer-like architectures have redefined the landscape of protein structure prediction.

1.1.4 Notable Works

In addition to AlphaFold2's profound impact on protein structure prediction, several other noteworthy contributions inform the research presented in this dissertation. For a more thorough introduction to these works, see the corresponding sections in Aim 1, Aim 2, and Aim 3.

That said, two works, in particular, have motivated this dissertation research and helped it evolve over time.

Focusing on Torsion Angle Prediction and Sidechain Modeling First, in February 2018, a few months into the beginning of my own work on protein structure prediction, Mohammed AlQuraishi of Harvard Medical School published a preprint article titled End-to-End Differentiable Learning of Protein Structure[18], in which he utilized RNNs to predict the structure of proteins by converting amino acid sequences to sequences of torsional angles. The torsional angles were used to reconstruct the protein, and a differentiable loss function called Distance-based Root Mean Squared Deviation (DRMSD) was used to compare the true and predicted structures and train the model. In contrast to Root Mean Squared Deviation (RMSD) which measures the inter-molecular distances $d(A_i, B_i)$ between all points in protein $A : (A_1...A_N)$ and their corresponding points in protein $B : (B_1...B_N)$ after alignment,

$$\mathbf{RMSD}(A, B) = \frac{1}{N} \cdot \sum_{i=1}^{N} d(A_i, B_i)$$

DRMSD first measures all intra-molecular distances between all points within each protein and then compares these two sets of distances:

$$\mathbf{DRMSD}(A, B) = \frac{1}{N^2 - N} \cdot \sum_{\forall i,j \in 1..N, i \neq j} d(d(A_i, A_j), d(B_i, B_j))$$

DRMSD effectively measures the similarity between two structures without having to align them first, making it completely differentiable and suitable for machine learning methods.

AlQuraishi's method proved effective as the first "end-to-end" method that could directly predict and optimize the DRMSD of a protein structure, though it encountered two specific challenges. First, the predicted torsional angles only considered the protein backbone and omitted pertinent information about the amino acid sidechains. Second, although the overall DRMSD of AlQuraishi's structures was low when submitting them to the CASP13 competition, due to issues with the predicted angles themselves, the structures were rejected automatically from the competition until the angles were corrected[19].

In late 2017, when work on this dissertation had just begun, our research group was developing a method very similar to AlQuraishi's. With the publishing of his method, we pivoted to focus on its limitations, mainly the lack of sidechain modeling.

Choosing the Right Architecture A second paper, this time in 2019 from Rao, Bhattacharya, and Thomas et al.[20], informed our research direction. The authors compared RNNs, Transformers, and a sequence alignment baseline on several tasks related to protein sequences[20]. These included the prediction of secondary structure, inter-molecular contacts, as well as protein engineering properties like fluorescence. The authors showed that, for some tasks, the performance of Transformer-based models exceeded other methods, while other tasks were best completed by RNN or alignment-based methods. The authors concluded that while many of the best models include sequence alignment information, further model development is needed to demonstrate the categorical superiority of any one method over another.

Despite the uncertainty in suggested model architectures suggested by the previous paper's authors, the Transformer model architecture regularly outperforms other methods on NMT tasks[14,15]. In addition, because the computational complexity of Transformer models is less than that of RNNs, they can be trained significantly faster and with a dramatically reduced training cost. In the original paper describing the Transformer model architecture, the proposed model outperformed state-of-the-art models with a 75% reduction in Floating-point Operations Per second (FLOPs) and training time on the order of days instead of weeks or months, which was previously considered standard for many NMT tasks[14].

With the progress and insights from the works mentioned above and recognizing the field's current needs, this dissertation has been framed to address several critical aspects and investigate the benefits of different architectures in different contexts.

1.2 Dissertation Overview

The trajectory of this dissertation has evolved in tandem with the rapid advancements in computational biology. In late 2017, our initial aspiration was straightforward: leveraging RNNs to predict protein structures from single amino acid sequences. However, with AlQuraishi's similar method and the rise of Transformers in sequence modeling, we had to rethink our approach.

We identified two main adjustments: (1) the need to model amino acid sidechains, which AlQuraishi's method missed; and (2) using Transformers, which have shown tremendous promise in sequence modeling. At the same time, we noticed a consistent issue in the community: a lack of accessible protein structure datasets for machine learning. This observation led us to focus on creating such a dataset in **Aim 1** while also working on our prediction models.

AlphaFold's success at CASP, along with the increased utilization of Multiple Sequence

Alignments (MSAs) in related methods, led us to re-evaluate our method-in-progress, the ProteinTransformer, which was not competitive with AlphaFold and did not utilize MSAs. However, our approach had a unique aspect: we could model all-atom protein structures end-to-end and use them directly in downstream molecular dynamics software. This inspired the core idea of **Aim 2**: harnessing the energy from predicted structures during training to refine the physical realism of the model. Taking what we learned, **Aim 3** ventures to improve AF2's sidechain modeling, which researchers have pointed out as a potential shortcoming.

In summary, this dissertation aims to combine deep learning and protein structure prediction to address existing challenges in datasets and methods. Our goal is threefold: improve the accessibility of protein structure data for machine learning, merge physical principles with advanced machine learning techniques, and achieve precision in all-atom prediction. Through this, we seek to bridge the gap between data-driven machine learning and the complex world of biological sciences.

2.0 SidechainNet: An All-Atom Protein Structure Dataset for Machine Learning

This chapter is adapted from:

> King, J. E., Koes, D. R. SidechainNet: An All-Atom Protein Structure Dataset for Machine Learning. *arXiv:2010.08162* **2020**. DOI: 10.48550/arXiv.2010.08162

which is also published as:

> King, J. E., Koes, D. R. SidechainNet: An all-atom protein structure dataset for machine learning. 2021; 89(11): 1489-1496. *Proteins* **2021**. DOI: 10.1002/prot.26169.

D. R. K. conceived the original project goal of sequence-based protein structure prediction via torsional angles. J. E. K. developed the original methods and datasets that were the first step toward this larger goal. D. R. K. and J. E. K. decided together to publish the SidechainNet dataset and methods. J. E. K. was the primary code author, carried out the analysis, generated figures, and wrote the manuscript. D. R. K. provided direction throughout.

2.1 Summary

Despite recent advancements in deep learning methods for protein structure prediction and representation, little focus has been directed at the simultaneous inclusion and prediction of protein backbone and sidechain structure information. We present SidechainNet, a new dataset that directly extends the ProteinNet dataset. SidechainNet includes angle and atomic coordinate information capable of describing all heavy atoms of each protein structure and

can be extended by users to include new protein structures as they are released. In this paper, we provide background information on the availability of protein structure data and the significance of ProteinNet. Thereafter, we argue for the potentially beneficial inclusion of sidechain information through SidechainNet, describe the process by which we organize SidechainNet, and provide a software package (`https://github.com/jonathanking/sidechainnet`) for data manipulation and training with machine learning models.

2.2 Introduction

The deep learning subfield of protein structure prediction and protein science has made considerable progress in the last several years. In addition to the success of deep learning methods in many of the most recent Critical Assessment of protein Structure Prediction (CASP) competitions[7,21? –23], many novel deep learning-based methods have been developed for protein representation[24,25], property prediction[26–28], and structure prediction[18,29–34]. Such methods are demonstrably effective and extremely promising for future research. They have the potential to make complex inferences about proteins much faster than competing computational methods and at a cost several orders of magnitude lower than experimental methods.

2.2.1 Protein Structure Data Availability and Information Leakage

The availability of existing data is a notable challenge for applications of deep learning methods to protein science and, in particular, to protein structure prediction. This limitation is not equally present in other applications of deep learning. For instance, in the field of image recognition, the ImageNet[8] dataset contains 14 million annotated and uniformly presented images. Similarly, in the field of natural language processing, linguistic databases and the web provide access to hundreds of millions of samples of annotated textual data and effectively limitless access to unannotated data. Though the Universal Protein Resource (UniProt) database as of September 2023 contains more than 250 million unique protein sequences[3], there are currently only 210,180 protein structures in the Protein Data Bank[4] (PDB), many of which are redundant in their structure or sequence information. Furthermore, although protein structure data is accessible through the PDB in individual files, it is not otherwise preprocessed for machine learning tasks in a manner analogous to other machine learning domains.

In addition to data availability, biased or skewed data presents another major challenge for machine learning practitioners[35]. Something as simple as dividing the data into training and testing splits for model development and evaluation can introduce harmful biases. Failure to analyze the similarities or differences between training and evaluation splits can lead to "information leakage" in which trained models exhibit overly optimistic performance by learning non-generalizable features from the training set. One such example of this in structural biology research is the Database of Useful Decoys: Enhanced (DUD-E) dataset[36]. DUD-E remains a common benchmark for evaluating molecular docking programs and chemoinformatic-focused machine learning methods despite recent research[37,38] uncovering dataset bias that explains misleading performance by many of the deep learning methods trained on it.

2.2.2 Treatment of Protein Backbone and Sidechain Information

A common way to describe the structure of a protein is to divide it into two separate components–the backbone and the sidechains that extend from it. The protein backbone is a linear chain of nitrogen, carbon, and oxygen atoms. The torsional angles (Φ, Ψ, and Ω) that connect these atoms form the overall shape of the protein. In contrast, protein sidechains are chemical groups of zero to ten heavy atoms connected to the central α-carbon of each amino acid residue. Each of the twenty distinct amino acids is defined by the unique structure and chemical composition of its sidechain component. Consequently, each protein is defined by the unique sequence of its constituent amino acids. The precise orientation of amino acid sidechains is critical to the biochemical function of proteins. Enzyme catalysis, drug binding, and protein-protein interactions all depend on a level of atomic precision that is not accounted for in backbone structure alone. Thus, the protein backbone and sidechain are both crucially important to protein structure and function.

Despite this, many predictive methods treat backbone structure prediction and complete sidechain structure prediction as distinct problems. For instance, one common approach is to predict the protein backbone alone and then add sidechains to the generated structure through a conformational or energy-minimizing search[39]. Another common method is to optimize the structure using backbone dependent rotamer libraries[40]. In both cases, sidechain placement is dependent on predicted backbone structure.

It is worth noting that some programs, like Rosetta[41], allow for the simultaneous optimization of protein backbone and sidechains. However, such methodology is not consistently adopted throughout the protein structure prediction literature. Relatedly, one notable result from CASP14, the most recent iteration of the CASP competition, was the success of AlphaFold 2[42], a deep learning protein structure prediction method. AlphaFold 2 incorporates, to some extent, both protein sidechain and backbone information in its procedure.

Still, the methods frequently employed by the research community that asynchronously predict protein backbone and sidechain structures remain effective. A possible reason for separating backbone and sidechain information could be that the community believes adequate progress in backbone structure prediction has been made without accounting for sidechain conformations during training. Alternatively, it may be the case that such information is simply not readily available for training deep learning models. Nonetheless, the potentially positive effect of utilizing the complete backbone and sidechain structure at training time has not been adequately addressed.

2.3 Methods

We propose SidechainNet, a protein sequence and structure dataset that addresses the concerns described in Sections 2.2.1 and 2.2.2. Namely, this work aims to account for data similarity across training splits, to improve upon existing datasets by adding features relevant to all-atom protein structure prediction, and, finally, to make such data available and easily accessible to researchers in both the machine learning and structural biology fields.

2.3.1 Addressing Protein Structure Data Availability and Information Leakage

SidechainNet is based on Mohammed AlQuraishi's ProteinNet[43]. ProteinNet is a dataset designed to mimic the assessment methodology of CASP proceedings.

Under the CASP contest organization, participants develop predictive methods using any publicly available protein structures. Thereafter, predictive methods are assessed using a specific set of proteins selected by contest organizers and withheld from participants until assessment. Organizers select proteins that are challenging to predict and, in the case of the Free Modeling contest category, have minimal similarity to available protein structures. In effect, the CASP process results in a portion of data for method development and a separate portion for evaluation. AlQuraishi proposed that these subsets could be treated as training and testing sets and re-purposed for machine learning.

Since no validation sets are constructed by the CASP organizers, AlQuraishi used clustering methods to extract distinct protein sequences and structures from each training set to use for validation. For each CASP contest, AlQuraishi ultimately constructed seven different validation sets–each containing an increasing upper bound on the similarity between sequences in the given validation set and those in the training set. AlQuraishi's construction allows users to evaluate a model's performance on proteins that have high similarity to a given training set (akin to the Template-Based Modeling CASP category) or proteins that have low or zero similarity to a training set (akin to the Free Modeling CASP category).

As a result of the constraints imposed by the CASP contest structure and AlQuraishi's validation set construction, ProteinNet accounts for protein sequence similarity across data splits. This prevents information leakage and minimizes misleading performance. For these reasons, we have replicated AlQuraishi's training, testing, and validation set constructions in SidechainNet.

2.3.2 Unifying Protein Backbone and Sidechain Information

SidechainNet also extends ProteinNet by including all heavy atoms of each protein and all necessary torsional angles for atomic reconstruction. We are interested in determining whether the availability of complete sidechain conformation information can improve the performance of structure prediction methods. However, even if such models are not more accurate, they will be fundamentally more expressive as they will generate all-atom protein structures. These structures could be used without modification in downstream analyses for structure-based drug discovery and even molecular dynamics simulations.

2.3.3 SidechainNet Properties

To the maximum extent possible, SidechainNet replicates the features included in ProteinNet (Table 2.1). However, SidechainNet also adds support for several new features. More information about the new features is included below.

Angle Information One important component of protein structure data often utilized in structure prediction methods but absent from ProteinNet is the set of torsional angles that describe the orientation of the protein. The canonical backbone torsional angles (Φ, Ψ, and Ω) for each residue are provided in SidechainNet.

Table 2.1: Differences between ProteinNet and SidechainNet. L represents protein length. [†]Position Specific Scoring Matrices (PSSMs) developed from multiple sequence alignments. [‡]SidechainNet includes atomic coordinates for backbone oxygen atoms while ProteinNet does not. [§]During processing, a select number of non-standard residues are converted to their standard forms (e.g., selenomethionine is partially converted to methionine). The unmodified primary sequence entry retains the 3-letter amino acid codes for each of the original amino acids in the sequence, while the primary sequence data entry contains the standardized 1-letter amino acid codes.

Entry	**Dimensionality**	**ProteinNet**	**SidechainNet**
Primary sequence	$L \times 1$	X	X
PSSM[†] + Information Content	$L \times 21$	X	X
Missing residue mask	$L \times 1$	X	X
Secondary structure labels	$L \times 1$	X	X
Backbone coordinates[‡]	$L \times 4 \times 3$	X	X
Backbone torsion angles	$L \times 3$		X
Backbone bond angles	$L \times 3$		X
Sidechain torsion angles	$L \times 6$		X
Sidechain coordinates	$L \times 10 \times 3$		X
Crystallographic resolution	1		X
Unmodified primary sequence[§]	$L \times 1$		X

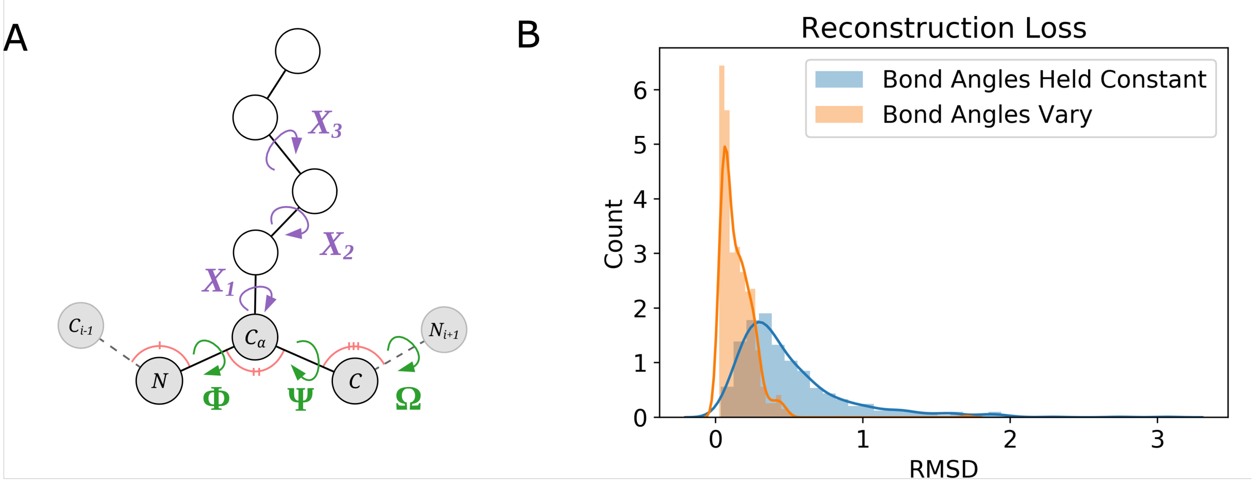


Figure 2.1: **(A)** A methionine residue annotated with the 3 categories of angles measured by SidechainNet: the canonical backbone torsional angles (Φ, Ψ, and Ω) in green, the rotatable sidechain-specific torsional angles (X_{1-3}) in purple, and the 3-atom backbone bond angles in pink. Atoms considered part of the residue's sidechain are colored white, while the backbone atoms are colored gray.

(B) Backbone bond angles (`C-N-CA`, `N-CA-C`, and `CA-C-N`; colored pink in 2.1**A**) are important for accurately reconstructing atomic coordinates from angles. When we assumed that these angles could be fixed during coordinate reconstruction, the result was greater reconstruction error (blue distribution). When we utilized the actual angle values during coordinate reconstruction, we achieved lower reconstruction error (orange distribution). Root-Mean-Square Deviation (RMSD) measures the difference between true atomic coordinates and coordinates reconstructed from recorded angles.

In the development of SidechainNet, we were surprised to observe that we incurred a noticeable amount of error when attempting to reconstruct a protein's Cartesian coordinates given only its backbone torsional angles (Figure 2.1B). We determined that the error was due to our assumption that several important non-torsional angles in the protein backbone (`C-N-CA`, `N-CA-C`, and `CA-C-N`) could be fixed using reference values from AMBER force fields[44] rather than being allowed to vary. Thus, even if a model perfectly predicted backbone torsional angles, it would still be subject to reconstruction error if such bond angles were held fixed. For that reason, we decided to include these angles in SidechainNet as well (Figures 2.1A and 2.1B).

Sidechain Information SidechainNet also includes sidechain-specific information that describes the orientation of each amino acid using both external (Cartesian) and internal (angular) coordinate systems. This information includes up to ten atomic coordinates and up to six torsional angles for the sidechain component of each residue.

The number of angles measured for an amino acid sidechain depends on the number of rotatable bonds within it. For residues with aromatic sidechains (e.g., tryptophan), the torsional angles describing the ring components of the residue are omitted since they can be inferred, but their complete atomic coordinates are still recorded.

3-atom bond angles (as opposed to 4-atom torsional angles) and bond lengths associated with sidechain structures are not recorded from structure files. Instead, we extracted reference values unique to each bond from the AMBER `ff19SB` force field[44,45]. When utilizing fixed angle values for sidechain 3-atom bond angles, the impact on reconstruction error is much smaller than the impact discussed in Figure 2.1 because the error does not accumulate between adjacent residues. Together with the recorded sidechain torsional angles, these angles and bond lengths can be used to generate all-atom sidechain structures.

2.3.4 Generation Procedure and Caveats

We constructed SidechainNet by, first, obtaining raw ProteinNet text records linked to in ProteinNet's GitHub repository (`https://github.com/aqlaboratory/proteinnet`). Then, we parsed these records before saving them into Python dictionaries. Next, we re-downloaded all-atom protein structures (sequences, coordinates, and angles) for each protein described by ProteinNet from the PDB using the ProDy software package[46]. Finally, we combined the remaining data (PSSMs, Information Content, and secondary structure labels) into the final SidechainNet dataset by aligning the sequences observed during the re-downloading process with the sequences described by ProteinNet. We replaced the coordinate and missing residue information described in ProteinNet with the re-downloaded values to ensure consistency. Scripts and instructions for constructing SidechainNet data are available in our GitHub repository.

We did not include data for 468 (0.4%) of the 104,323 listed protein structures from ProteinNet's CASP12 dataset. Some of these structures are of proteins that include D-amino acids. Such structures were purposely excluded despite being included in ProteinNet because they cannot be effectively modeled with L-amino acid sidechain structures. Other structures were excluded due to problematic file parsing, edge cases, and disagreement between the sequences we observed and those reported by ProteinNet that could not be resolved programmatically.

Furthermore, both ProteinNet and SidechainNet only consider single-molecule protein chains. Proteins from the PDB that contain multiple chains are divided into multiple independent entries, one for each chain. Any missing residue or atomic information is padded with vectors that contain `sidechainnet.GLOBAL_PAD_CHAR` and match the size of the corresponding data (e.g., 1×3 for each missing atom, 1×1 for each missing angle). In addition, detailed characteristics of structure data (e.g., b-factors, multiple or alternative sidechain locations) are ignored.

Lastly, structures in SidechainNet are recorded directly from the PDB and have no guarantee to be energetically minimized. In Aim 2, we develop a version of SidechainNet that is minimized using AMBER force fields and the OpenMM software package to address this shortcoming. This may make the data more consistent and amenable to training.

2.4 Results

SidechainNet was originally developed for Python and the PyTorch machine learning framework due to their ease of use and popularity within the research community. Here we discuss how researchers may use our Python package to load SidechainNet and efficiently train models with it.

2.4.1 SidechainNet as a Python dictionary

In its simplest form, SidechainNet is stored as a Python dictionary organized by the same training, validation, and testing splits described in ProteinNet. There are multiple validation sets included therein.

Within each of SidechainNet's training, validation, and testing splits is another dictionary mapping data types (`seq`, `ang`, etc.) to a list containing this data for every protein. Shown below, `seq{i}`, `ang{i}`, etc. refer to the `i`th protein in the dataset.

```
data = {"train": {"seq": [seq1, seq2, ...],  # Sequences, 1-letter codes
                  "ang": [ang1, ang2, ...],  # Angles
                  "crd": [crd1, crd2, ...],  # Coordinates
                  "msk": [msk1, msk2, ...],  # Missing residue masks
                  "evo": [evo1, evo2, ...],  # PSSMs and Information Content
                  "sec": [sec1, sec2, ...],  # Secondary structure labels
                  "res": [res1, res2,  ...], # X-ray crystallographic resolution
```

```
          "mod": [mod1, mod2, ...],  # Modified residue annotations
          "ums": [ums1, ums2, ...]   # Unmodified sequences, 3-letter codes
          "ids": [id1,  id2,  ...],  # Corresponding ProteinNet IDs
          },
"valid-10": {...},
    ...
"valid-90": {...},
"test":     {...}
}
```

By default, the load function downloads the data from the web into the current directory and loads it as a Python dictionary. If the data already exists locally, it reads it from disk. Other than the requirement that the data must be loaded using Python, this method of data loading is agnostic to any downstream analyses.

```
>>> import sidechainnet as scn
>>> data = scn.load(casp_version=12)
```

2.4.2 SidechainNet as PyTorch DataLoaders

The load function can also be used to load SidechainNet data as a dictionary of torch.utils.data.DataLoad objects. PyTorch DataLoaders make it simple to iterate over dataset items for training machine learning models. This method is recommended for using SidechainNet data with PyTorch. For a complete description of the data available to the user when using SidechainNet's DataLoaders, see Table 2.2.

By default, the provided DataLoaders use a custom batching method that randomly generates batches of proteins of similar length. For efficient GPU usage, it generates larger batches when the average length of proteins in the batch is small and smaller batches when the proteins are large. The probability of selecting small-length batches is decreased so that each protein in SidechainNet is included in a batch with equal probability. See dynamic_batch_size and collate_fn arguments for more information on modifying this behavior.

```
>>> dataloaders = scn.load(casp_version=12, with_pytorch="dataloaders")
>>> for batch in dataloaders['train']:
....    predicted_angles = model(batch.seqs)
....    sb = scn.BatchedStructureBuilder(batch.int_seqs, predicted_angles)
....    predicted_coords = sb.build()
....    # Compute loss between any group of true and predicted values
....    loss = compute_loss(batch.angs, batch.crds,
....                        predicted_angles, predicted_coords)
....    ...
```

Table 2.2: Data attributes available to the user when utilizing SidechainNet's custom DataLoaders. †May be formatted as integer or one-hot sequences depending on value of scn.load(...seq_as_onehot=...).

Batch Attribute	Description
pids	Tuple of ProteinNet/SidechainNet IDs
seqs	Tensor of sequences†
int_seqs	Tensor of sequences represented as integers
str_seqs	Tuple of sequences represented as strings
msks	Tensor of missing residue masks (redundant with padding in data)
evos	Tensor of Position Specific Scoring Matrix + Information Content
secs	Tensor of secondary structure labels†
angs	Tensor of angles
crds	Tensor of coordinates
seq_evo_sec	Tensor with concatenated values of seqs, evos, and secs
resolutions	Tuple of crystallographic resolutions (Å), when available
is_modified	Tensor of modified residue bit-vectors. Each entry is a bit-vector where a 1 signifies that the residue at that position has been modified to match a standard residue supported by SidechainNet (e.g., selenomethionine to methionine)

2.4.3 Converting Angle Representations into All-Atom Structures

An important component of this work is the inclusion of both angular and 3D coordinate representations of each protein. Researchers developing methods that rely on angular representations may be interested in converting this information into Cartesian coordinates. For this reason, SidechainNet provides tools to convert angles into coordinates.

In the below example, `angles` is a NumPy matrix or Torch Tensor following the same organization as the NumPy angle matrices provided in SidechainNet. `sequence` is a string representing the protein's amino acid sequence. Both of these are obtainable from the SidechainNet data structures described in Sections 2.4.1 and 2.4.2.

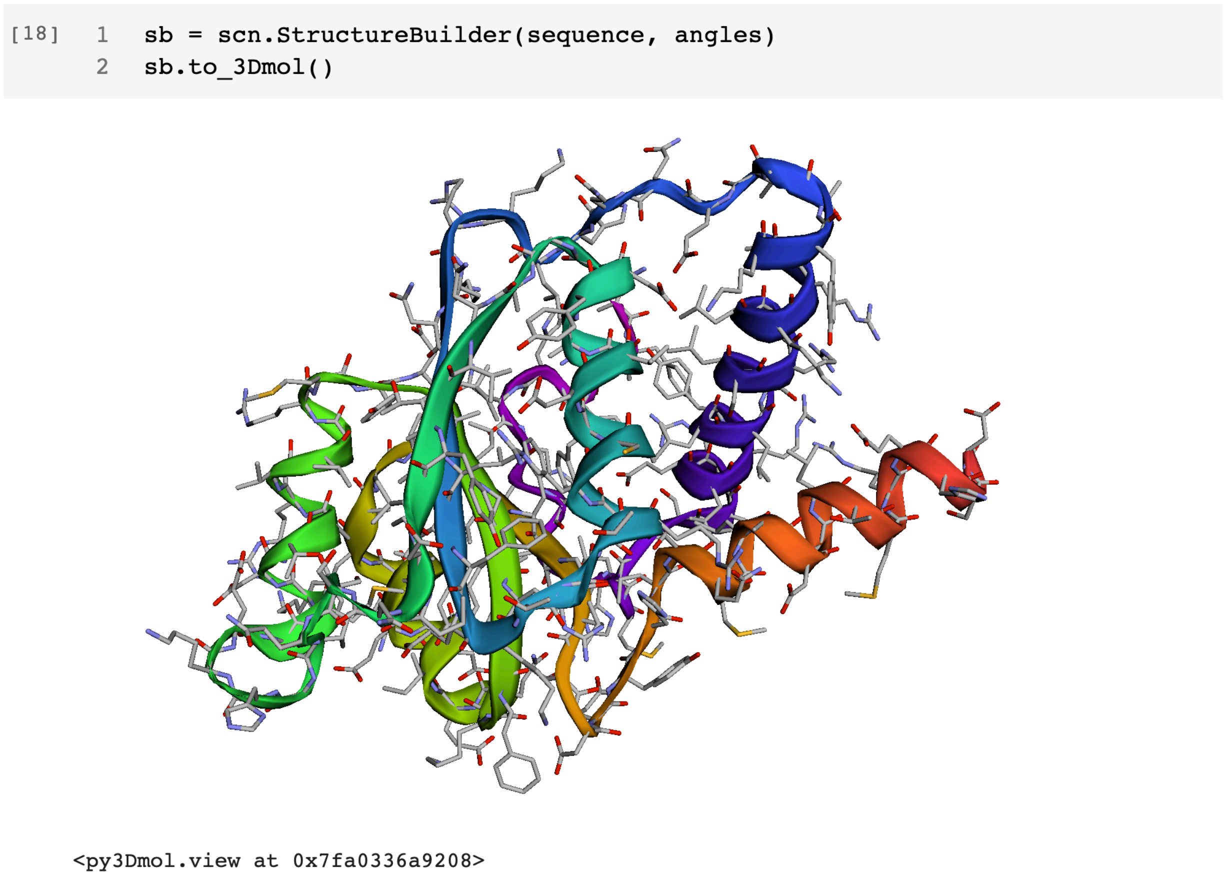
```
>>> sequence, angles = data['train']['seq'][0], data['train']['ang'][0]
>>> (len(sequence), angles.shape)  # 12 angles per residue
(128, (128, 12))
>>> sb = scn.StructureBuilder(sequence, angles)
>>> coords = sb.build()
>>> coords.shape  # 14 atoms per residue (128*14 = 1792)
(1792, 3)
```

2.4.4 Visualizing All-Atom Structures with PDB, py3Dmol, and gLTF Formats

SidechainNet makes it easy to visualize both existing and predicted all-atom protein structures, as well as structures represented as either angles or coordinates. These visualizations are available as PDB files, `py3Dmol.view` objects, and Graphics Library Transmission Format (gLTF) files. Examples of each are included below.

The PDB format is a typical format for representing protein structures and can be opened in software tools like PyMOL[47]. `py3Dmol` (built on 3Dmol.js[48]) enables users to visualize and interact with protein structures on the web and in Jupyter Notebooks via an open-source, object-oriented, and hardware-accelerated Javascript library (Figure 2.2). Finally, gLTF files, despite their larger size, can be convenient for visualizing proteins on the web or in contexts where other protein visualization tools are not supported.

```
>>> sb.to_pdb("example.pdb")
>>> sb.to_gltf("example.gltf")
```



```
[18]  sb = scn.StructureBuilder(sequence, angles)
      sb.to_3Dmol()
```

```
<py3Dmol.view at 0x7fa0336a9208>
```

Figure 2.2: `py3Dmol` enables the user to interactively inspect all-atom protein structures from SidechainNet within Jupyter/iPython Notebooks.

2.4.5 Extending SidechainNet with User-Specified Data

Finally, we have considered the fact that users may be interested in utilizing SidechainNet's functionality without strictly conforming to SidechainNet's original organization and composition. To aid researchers with potentially different goals than the authors, we allow users to

(1) specify which proteins to include in SidechainNet, and/or (2) customize their data split assignment. This functionality, found in `sidechainnet. create_custom`, enables users to work around the predetermined list of proteins and dataset splits inherited from ProteinNet. Importantly, this also allows SidechainNet to expand to include new protein structures as they are released. That said, information directly acquired from ProteinNet (e.g., Position Specific Scoring Matrices and secondary structure information) may be omitted if users' specified proteins were not previously included in ProteinNet.

2.5 Discussion and Conclusion

SidechainNet builds upon the strong foundation of ProteinNet to provide a protein structure dataset with minimal bias and maximum information content. By including many of the angles necessary to accurately reconstruct atomic Cartesian coordinates from an angle-only protein representation, we also enable future users to develop methods that utilize either or both internal and external coordinate systems with a previously inaccessible level of atomic detail and model complexity. In addition, the tools developed for SidechainNet's programmatic construction may also be adapted for the generation of similar all-atom datasets with alternative methods of data clustering (e.g., datasets based on CATH[49]).

Methods developed to explicitly include the orientation and atomic coordinates of protein sidechains may have the upper hand in tasks such as structure-based drug discovery or in the analysis of specific enzyme activities, where such information is necessary. Although the impact of including sidechain information on existing predictive methods has yet to be studied, we hope SidechainNet makes such research more accessible.

As we had hoped, since the time this work was originally published, SidechainNet has been adopted by the community and has been used in various research projects. At the time of writing this dissertation, SidechainNet has been starred For example, in the year or so preceeding the eventual publication of AlphaFold, many code repositories on GitHub were created in an attempt to reverse-engineer it. The most popular open source repository, by far, Additionally, SidechainNet may find use in applications beyond protein structure prediction by making highly organized and uniformly preprocessed protein structure and sequence data easily accessible via Python. Protein science is currently experiencing a critical moment with respect to the development of new machine learning methods. The data that SidechainNet provides may prove to be a useful tool for developing methods to study protein representation, predict protein structure-property relationships, or to predict protein-protein and protein-ligand interactions.

Since the time this work was originally published, SidechainNet has been adopted by the community and has been used in various research projects. At the time of writing this dissertation, SidechainNet has been "starred" (i.e., bookmarked) on GitHub 283 times and "forked" (i.e., duplicated for modification) 32 times. See `https://github.com/jonathanking/sidechainnet/network/dependents` for a summary of projects available on GitHub that utilize Sidechainnet. Some notable examples of SidechainNet's use by the community include:

- An implementation of "Denoising Diffusion Probabilistic Model for Proteins"[50], originally written by Ho et al., but implemented by user lucidrains. [GitHub]
- An implementation of "Single-sequence protein structure prediction using language models from deep learning"[33], a.k.a. "RGN2", originally written by Chowdhury et al. but implemented by user hypnopump. [GitHub]
- Original implementation of "Coarse-Graining Variational Autoencoders(CGVAE)"[51], [GitHub]
- ProteinFlow computational pipeline, a product from AdaptyvBio, a startup company. [GitHub]
- "MP-NeRF: Massively Parallel Natural Extension of Reference Frame"[52]. [GitHub]

- A popular open-source implementation (1,404 stars) of AlphaFold2 (created when it was not clear if AF2 would ever be published), also implemented by lucidrains. [GitHub]
- ClynMut: Predicting the Clinical Relevance of Genome Mutations. [GitHub]
- "Contextual protein and antibody encodings from equivariant graph transformers"[53]
- "Ultra-fast protein structure prediction to capture effects of sequence variation in mutation movies"[54]
- "Protein Function Analysis through Machine Learning"[55]

and others.

Protein science is currently experiencing a critical moment with respect to the development of new machine learning methods. The data and framework that SidechainNet provides has proven to be a useful tool for developing methods to study protein representation, predict protein structure-property relationships, or to predict protein-protein and protein-ligand interactions.

Source code and example machine learning workflows using SidechainNet can be found at `https://github.com/jonathanking/sidechainnet`. To install, use the `pip` command-line program (`pip install sidechainnet`).

3.0 Interpreting Molecular Dynamic Forces as Deep Learning Gradients Improves Quality Of Predicted Protein Structures

This chapter is under review for publication at the Biophysical Journal.

D. R. K. conceived the project and developed a parallelized method for all-heavy-atom structure generation of atomic coordinates from torsional angles. All other code, method development, and analyses were completed by J. E. K. The manuscript was written by J. E. K. with guidance provided by D. R. K.

3.1 Summary

Protein structure predictions from deep learning models like AlphaFold2, despite their remarkable accuracy, are likely insufficient for direct use in downstream tasks like molecular docking. The functionality of such models could be improved with a combination of increased accuracy and physical intuition. We propose a new method to train deep learning protein structure prediction models using molecular dynamics force fields to work toward these goals. Our custom PyTorch loss function, OpenMM-Loss, represents the potential energy of a predicted structure. OpenMM-Loss can be applied to any all-atom representation of protein structure capable of mapping into our software package, SidechainNet. We demonstrate our method's efficacy by finetuning OpenFold. We show that subsequently predicted protein structures, both before and after a relaxation procedure, exhibit comparable accuracy while displaying lower potential energy and improved structural quality as assessed by MolProbity

metrics.

3.2 Introduction

Deep learning (DL) methods for protein structure prediction, particularly AlphaFold2 (AF2)[6] and AF2-like systems[32,34,56], have proven remarkably accurate. However, barriers to the practical use of AF2 predictions remain. Some of AF2's shortcomings are due to limitations of its design, while others are simply areas where accuracy could be improved. For instance, Jumper et al.[6] trained AF2 on protein structures from the Protein Data Bank (PDB)[57]. By virtue of their presence in the PDB, many of these proteins formed stable crystal structures in the lab. Such structures may not adequately reflect their *in vivo* conformations, which may exhibit higher energy under crystallization conditions. Another fundamental limitation of AF2 is its inability to simultaneously model proteins and their ligands. Without knowledge of protein-ligand interactions, AF2 cannot reliably predict *apo* or *holo* protein states, making drug development with these structures more challenging.

AF2 also likely needs higher accuracy for downstream tasks like docking. Several studies[58–60] have demonstrated that AF2-predicted structures are surprisingly non-performant compared to crystal structures used for docking. Despite a relatively high level of accuracy of the protein backbone and binding sites, subtle differences between the prediction and crystal structure cause significantly different results when docking molecules. Sidechain atom accuracy is a likely culprit since differences in sidechain atom placement or orientation may dramatically alter the binding pocket shape or chemical environment.

In addition to working toward more accurate models, the research community has expressed interest in developing models capable of understanding and recapitulating fundamental physical principles. Models that leverage existing scientific knowledge could free up computational resources to focus on subproblems better suited for machine learning instead of relearning principles taught in high school chemistry and mathematics. Such a model could, in principle, better guide drug discovery campaigns since it may better understand the chemical interactions between a protein and a ligand.

Researchers in several fields have tried utilizing more complex and physically relevant input representations and model architectures or even empirical data to improve prediction quality. For example, data representation in cheminformatics has shifted from SMILES[61] (string-based) representations of chemical structures to other representations like graphs[62,63], point clouds[64], or 3D voxels[65] that more closely resemble the physical reality of these systems. One DL method for computational drug discovery utilizes theoretical chemical principles in a Siamese neural network[66]. By directly subtracting the latent spaces between the two component models, the authors model the relative free binding energy $\Delta\Delta G$ between two ligands, enforcing the symmetric properties of relative binding affinities $\Delta\Delta G_{AB} = \Delta G_A - \Delta G_B = -(\Delta G_B - \Delta G_A)$. Active learning is also often used to incorporate experimental data into machine learning training procedures, guiding machine learning experiments with real-life data[67–69].

In the same spirit, AF2's authors incorporate physiochemical constraints into their work by training with a "violation loss." The violation loss is a component of AF2's composite loss function that penalizes predictions for generating structures with overlapping (i.e., clashing) atoms or bond lengths and angles that deviate from the corresponding values in the literature. Despite the addition of the violation loss, AF2 predictions have significant stereochemical violations. The recommended prediction procedure involves a relaxation step with molecular dynamics force fields (MD FFs), which takes several seconds but significantly improves the final structure by removing many of these violations. Triangle point attention, also developed by AF2's authors, is suggested to provide physical and geometric intuition to AF2. However, because it is a neural network component with learned parameters, it is unclear how it works in practice or if it genuinely imbues the model with any physical constraints.

3.3 Methods

This section describes our method to improve structure predictions from methods like AF2 by incorporating physical principals from molecular dynamics force fields. We will discuss the intuition for our approach, practical considerations for its development, and our procedure for training and evaluating our models with custom datasets.

OpenFold is an implementation of AlphaFold2 that attempts to faithfully reproduce AF2 in PyTorch with added utilities for training. OpenFold was developed by Ahdritz et al.[70], and AlphaFold2 by Jumper et al.[6]. We refer to OpenFold and AlphaFold2 interchangeably as AF2, though we only work with the OpenFold code base.

3.3.1 Forces from molecular dynamics can be interpreted as gradients in backpropagation.

To improve overall protein structure prediction accuracy and to help protein structure prediction models better understand physical principles, we propose capitalizing on the relationship between the physical forces computed by molecular dynamics (MD) software (**Eq 3.1**) and the gradients utilized in machine learning backpropagation (**Eq 3.2**).

$$F(X) = -\nabla_X U(X) \tag{3.1}$$

MD software evaluates the energy $U(X)$ of a molecular system at each time step based on the positions of the atoms X and the force field parameters. Next, the software computes the forces acting on the atoms via **Eq 3.1**. Since MD software can readily compute $U(X)$ and $-\nabla_X U(X)$ for a protein system, we propose treating $U(X)$ as a loss function whose gradients are $-F(X)$.

$$
\begin{aligned}
\text{Loss} &= U(X) \\
\nabla_X(\text{Loss}) &= -F(X)
\end{aligned}
\tag{3.2}
$$

For backpropagation machinery to utilize these gradients during training, all that is needed is a custom layer with "forward" and "backward" components that return $U(X)$ and $-F(X)$, respectively. We can then apply this framework to any machine learning model capable of predicting all-atom protein structure representations.

3.3.2 OpenMM-based loss function

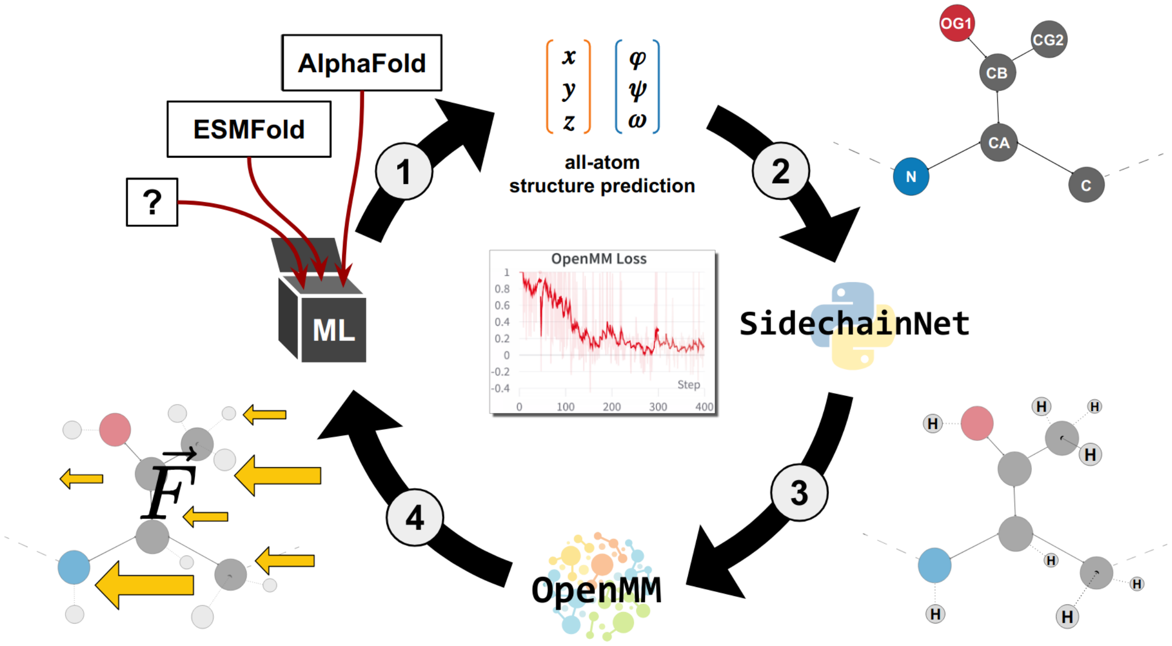


Figure 3.1: A graphical summary of our method for training deep learning protein structure prediction models by interpreting MD forces as gradients.

Our custom loss function implements **Eq 3.2** via the steps depicted in **Fig 3.1**. **(1)** A deep learning model predicts an all-atom protein structure representation. Our code supports Cartesian and internal coordinate (bond and torsional angle) representations of protein structure. **(2)** The user determines a mapping between the atoms or angles predicted by their model and the atom or angle representation expected by the SidechainNet protein representation developed in Aim 2. SidechainNet[71] is a Python library and dataset with tools for PyTorch-based[72] deep learning protein structure prediction tasks. **(3)** In preparation for energy evaluation with OpenMM[73], a molecular dynamics system is initialized using the all-atom protein representation, and hydrogen atoms are added (if not already present) to the prediction using differentiable vector operations. **(4)** The custom loss function layer and SidechainNet protein representation compute a loss equal to the predicted structure's potential energy. The forces acting on the atoms, also calculated by OpenMM, are designated as gradients to train the model. The model parameters are updated, and training continues.

Users may specify any force field and solvent model available in the OpenMM package for use with our method. For experiments in this paper, we selected the Generalized Born Implicit Solvent (`gbn`) model along with the `ff15ipq` force field[74].

Practical Consideration 1: Hydrogens Let X_H and $X_{\bar{H}}$ represent a predicted protein structure with and without hydrogen atoms. MD FFs require protein systems to contain hydrogen atoms in order to compute $U(X_H)$ and $F(X_H)$. However, AF2 and related models predict structures without hydrogens, $X_{\bar{H}}$. Thus, we must find a way to add hydrogens to the structures predicted by these models during training so that we may utilize the values computed by OpenMM. If hydrogens are added to the predicted structure in a non-differentiable manner (e.g., via OpenMM), we cannot train our model end-to-end. This is because we must be able to compute the gradients or construct a forward and backward component for every layer of our neural network. Adding hydrogens outside of PyTorch would not retain the computation graph necessary for automatic differentiation. In other words, we would not easily be able to compute the derivative of the function, g, that adds hydrogens to the predicted structure, $g(X_{\bar{H}}) = X_H$.

We developed software in the SidechainNet package to address this issue. Our method adds hydrogens (if needed) to predicted protein structures, using only differentiable PyTorch vector operations to maintain differentiability. The software components found in SidechainNet's `SCNProtein` class enable building hydrogen-inclusive Cartesian coordinates starting from angle-only representations via a sidechain-parallel method based on Natural extension of Reference Frame (NeRF)[75] or by adding hydrogens to the Cartesian coordinates of existing heavy atoms, also via NeRF. Next, our software sets up an OpenMM MD system. Our custom loss function is then used to compute $U(X_H)$ in the forward pass and $-F(X_H)$ in the backward pass by querying these values from OpenMM. Since converting $X_{\bar{H}} \rightarrow X_H$ via vector operations is differentiable, PyTorch can effectively map the all-atom forces back to the heavy atom coordinate representation.

Our method adds hydrogens in consistent, predetermined conformations without further optimization. Although this placement may result in non-optimal hydrogen positions, this does not prevent our system from minimizing the structure's potential energy.

Practical Consideration 2: Rugged Potential Energy Landscapes Training a machine learning model using potential energy as a loss function is challenging because protein energy landscapes can be incredibly rugged[76,77]. Small changes in sidechain rotamers may cause unrealistic clashes in otherwise perfect structures. Suppose MD FFs evaluate the energy of a protein system with clashes. In that case, the software may return astronomically high energy values (infinite or approaching the maximum value of computational numerical representation). These values reflect the physical impossibility of two atoms occupying the same physical space. This behavior is not frequently observed in MD trajectories because atoms often do not move enough between time steps to overlap or take on unrealistic conformations. However, when training DL models using mini-batches for gradient descent, each subsequent prediction is for a different protein and represents a new MD system. Thus, there are no guarantees that the predicted system will be in a realistic conformation. Furthermore, in a model like AF2, there are several other loss terms in the composite loss function to consider (e.g., Frame Aligned Point Error (FAPE) loss, distogram loss, and masked multiple sequence alignment loss). The raw energy of the predicted system would easily dominate the magnitude of the other loss terms, which we observed having maximal values of about 1.5 loss units in a trained AF2 model.

To smooth out the energy landscape's ruggedness and transform the loss's range into a range that is less likely to dominate other loss terms, we construct a sigmoid function, $\tilde{\sigma}$, by modifying the standard sigmoid function, σ, and applying it directly to the raw energy, x (Figure 3.2).

Parameter	Value
A	5
B	10^6
C	$3 \cdot 10^5$
D	5

$$\begin{aligned} \sigma(x) &= (1 + e^{-x})^{-1} \\ \tilde{\sigma}(x) &= \left(\frac{1}{A} + e^{-\frac{(Dx+B)}{C}} \right)^{-1} + A + 1 \\ \sigma^*(x) &= \frac{x}{10^{12}} + \tilde{\sigma}(x) \end{aligned} \tag{3.3}$$

We heuristically chose several constant scalar parameters to construct a sigmoid function with desirable properties. $\tilde{\sigma}$ has a maximum value of 1, a minimum value of -4, and changes from positive to negative when the energy of a protein structure is about -20,000 kJ/mol. We also selected a loss component weight of 0.01 for the final computed loss value. These values were chosen so that the range of our loss function during training would not differ significantly from the range of values of the other loss components. For example, we reasoned that since AF2 loss components did not become negative during our finetuning experiments, we would like our energy-based loss to become negative only after surpassing a low-energy threshold of -20,000 kJ/mol that we associated with energetically favorable protein structures. Allowing our loss function to become negative would enable the model to optimize the predicted structure's energy further, even when other loss components reached their minimal values.

We developed a second modified sigmoid, σ^*, which is a "leaky" version of $\tilde{\sigma}$. σ^* adds an extremely small fraction of x to $\tilde{\sigma}$. This enables constant, non-zero gradients for even extreme energy loss values. Theoretically, this allows the model to learn from physically implausible structures while still dramatically smoothing the energy-loss landscape. This modification also removes the asymptotic lower bound in $\tilde{\sigma}$ so that models may reach any energetic minimum.

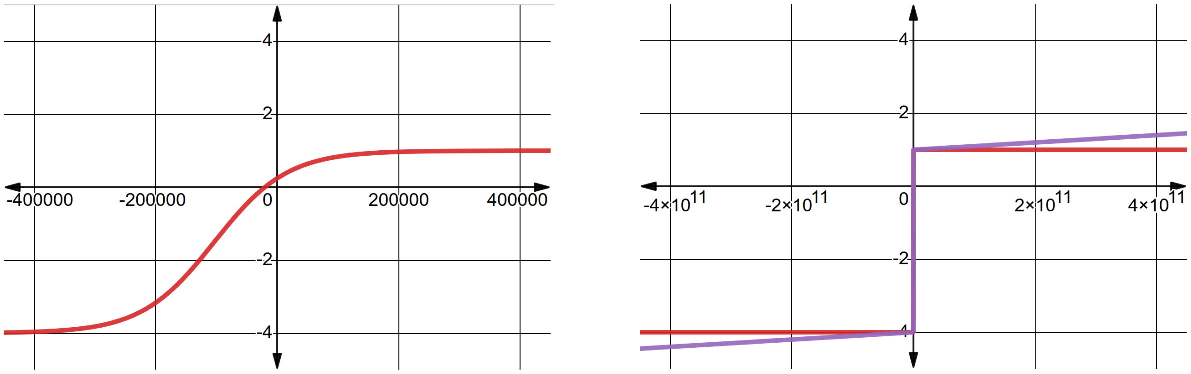


Figure 3.2: $\tilde{\sigma}$ and σ^* in red and purple, respectively. Both functions have nearly identical behavior (left), except σ^* has a non-zero derivative for extreme energy values (right).

In practice, we found these modifications helpful for training and visualizing energy values since the range of raw energy values of predictions can be too large to plot even on a log-scaled axis.

3.3.3 Training

Models To measure the effects of our custom loss function, we developed a fork of the OpenFold model that enables training with our data and loss function. We finetuned models rather than training from scratch, loading weights provided by the OpenFold authors.

We attempted to match the finetuning procedures described in the OpenFold and AF2 papers. However, we did not increase the Multiple Sequence Alignment (MSA) depth or crop size to limit GPU memory consumption. We trained each model using four A100s for about 12 days and achieved an effective batch size of 128 by accumulating gradients 32 times. To avoid training instability, we used an OpenMM-Loss-specific learning rate schedule that linearly scaled the loss function component weight to its maximum value over 1000 steps.

Data We propose two distinct and mutually compatible ways to train AF2 to predict structures with lower energy and better physical characteristics: training on energetically minimized protein structures and training with our loss function.

We propose training on energetically minimized protein structures for the following reason. Suppose we train AF2 on raw, unminimized PDB structures with our loss function. As mentioned before, structures in the PDB likely do not reflect realistic structures in an aqueous environment. In that case, the accuracy-based (e.g., FAPE) and energy-based (e.g., OpenMM-Loss) loss function components may push the model in opposing directions, one towards accurately reflecting the PDB structure and another towards predicting a conformation with low energy. Training on protein structures that already exist at a local minimum of our loss function should enable the model to satisfy both objectives simultaneously.

To create this minimized dataset, we used our loss function (without the sigmoid component) to minimize a set of proteins from the SidechainNet dataset. We chose to minimize the energy of our proteins with respect to their bond and torsional angles rather than their atomic coordinates. This theoretically allows models that predict angle representations of proteins to predict their target with perfect accuracy since bond lengths and angles in the minimized set are set to idealized values with respect to the `ff15ipq` force field. Starting with about 100,000 protein entries from the SidechainNet CASP 12-era[78] dataset, we ended up with about 32,000 entries that (1) did not have gaps in their structures (a requirement for our minimization procedure); and (2) did not deviate more than 5 RMSD from their starting structure after minimization. This dataset is called **scnmin**.

An unminimized version of this dataset containing identical protein entries, called **scnunmin**, was retained to evaluate our loss function without minimized protein structures.

We utilized MSAs and template hits generated by the OpenFold authors for training when available[70]. We followed the `mmseqs2`-based[79] procedure suggested by the OpenFold authors to generate the remaining input MSAs and template hits.

We chose a validation set from CAMEO[80] to match the one in the OpenFold paper.

We constructed a new test set containing 93 proteins from a one-year window of CAMEO proteins ending on January 3, 2023. Our validation and test sets were minimized similarly to the training set while retaining their unminimized counterparts. In cases where a protein was rejected from our minimization procedure, we excluded it from both the minimized and unminimized sets. To determine training convergence, we used the minimized version of the validation set when training with **scnmin** and the unminimized validation set when training with **scnunmin**.

3.3.4 Evaluation

It is essential to briefly discuss the distinction between accuracy (e.g., lDDT[81]) and structural quality (e.g., MolProbity Score). Methods papers for protein structure prediction have understandably focused on accurately predicting the protein structures available in the PDB. However, as discussed earlier, these structures are not ideal prediction objectives because they are potentially dissimilar from their *in vivo* counterparts. Additionally, their structures may change significantly in different environments or when bound to other ligands.

Accuracy metrics necessitate a selection of ground truth structures. Because we trained models to predict either minimized or unminimized structures from the PDB, using accuracy alone to assess our method's performance would be insufficient. Instead, we propose that a more appropriate assessment of our method is to compare orthogonal measures of structural quality—such as the MolProbity Score or the Clash Score computed by the MolProbity software suite[82]. The MolProbity Score for a given structure reflects the crystallographic resolution at which the observed amount of structural irregularities would be expected. For example, a MolProbity Score of 0.8 would be the expected score from a structure determined from an X-ray crystallographic experiment with a resolution of 0.8 Å. The Clash Score measures per-contact clash energy in a protein structure when atoms are unrealistically close to each other. In both cases, lower values are better. We assess structural quality metrics before and after AF2-prescribed relaxation with OpenMM and AMBER force fields.

We repeated each finetuning experiment with three different random seeds to measure the robustness of our method. In the case of AF2, we utilized three different sets of weights reported to have similar performance instead of retraining AF2 from scratch (`finetuning_3.pt`, `finetuning_4.pt`, and `finetuning_5.pt`). In the subsequent plots and analysis, we combined the results from all three seeds for simplicity. However, the trends we observe are consistent across all seeds.

We checkpointed models by their performance on the validation set with respect to the OpenMM-Loss. We computed p-values using the one-sided T-Test implemented in `scipy`[83] with respect to the AF2 baseline.

3.4 Results

3.4.1 OpenMM-Loss lowers prediction potential energy while maintaining accuracy.

When trained with our loss function, predictions exhibit significantly lower energy loss values compared to predictions made from AF2 (**Fig 3.3**). Although we do not plot raw energy values (kJ/mol) here, we argue OpenMM-Loss is an appropriate substitution because it monotonically increases with energy and simplifies the interpretation of extreme values. lDDT_{AA}, an all-atom accuracy metric, is not significantly different (p=0.8) for both methods when evaluated on the energy-minimized CAMEO test set (**Fig 3.4**). We use a simplified implementation of lDDT_{AA} that avoids renaming ambiguous atom names and thus slightly under-reports accuracy.

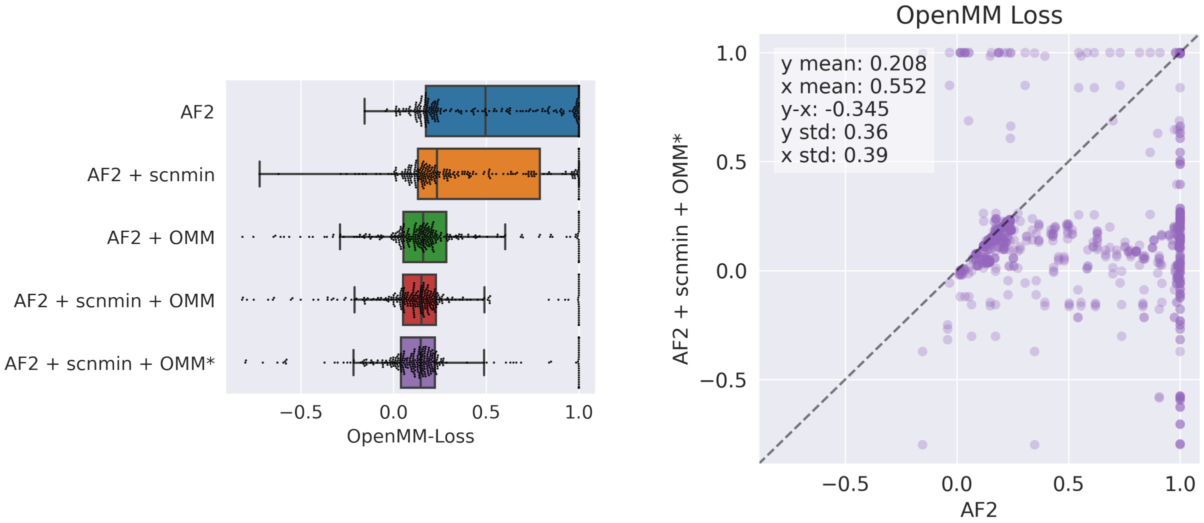


Figure 3.3: Model prediction potential energy represented by OpenMM-Loss. Each dot represents a predicted protein structure from the CAMEO test set before relaxation. **AF2**: baseline AlphaFold2 initialized using `finetuning_5.pt`. + **scnmin**: finetuned on minimized SidechainNet proteins instead of unminimized structures. + **OMM**: finetuned using OpenMM-Loss with $\tilde{\sigma}$ activation. + **OMM***: finetuned using OpenMM-Loss with σ^* activation.

3.4.2 OpenMM-Loss improves physical properties of predictions before and after relaxation.

Our method's predictions have more favorable MolProbity Scores and Clash Scores on average than predictions from AF2 (**Fig 3.5**, **Fig 3.6**, and **Table 3.1**). This is true both before and after the AF2 relaxation procedure, indicating that simply minimizing AF2 predictions is not as beneficial as incorporating energy terms into the training procedure. Our method also has significantly more predictions with the lowest possible values of either metric (0.5 and 0, respectively). We observe a trend from top to bottom in **Fig 3.5** of improving structure quality as we add the components of our method (training on minimized structures, **scnmin**, and training with our custom loss, **OMM/OMM***).

It is worth noting that training on minimized structures (**scnmin**) alone does not significantly improve accuracy or structural quality metrics despite lowering the potential energy

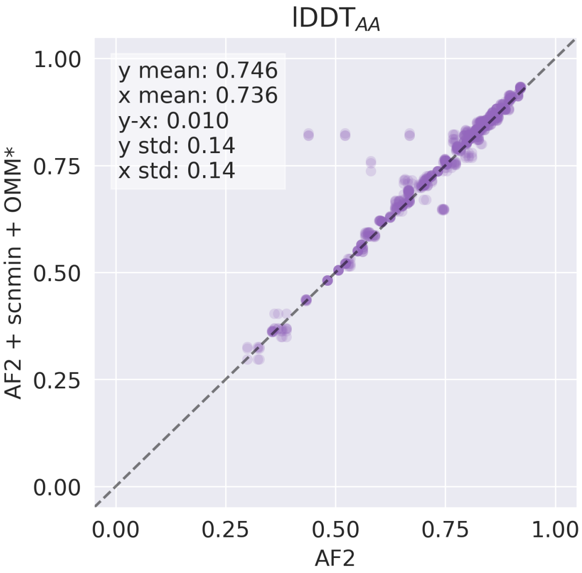


Figure 3.4: Accuracy between unrelaxed model predictions and the minimized CAMEO test when comparing our method and the **AF2** baseline. **AF2**: baseline AlphaFold2 initialized using `finetuning_5.pt`. + **scnmin**: finetuned on minimized SidechainNet proteins instead of unminimized structures. + **OMM***: finetuned using OpenMM-Loss with σ^* activation.

of predicted structures on average. In contrast, our loss function significantly improves structural quality relative to the **AF2** baseline. The best improvement is seen by combining these two training strategies.

The accuracy of our models between unrelaxed structures and the minimized CAMEO test set labels is shown in **Table 3.2**. Our objective is to produce more realistic predictions while not necessarily reproducing structures from the PDB, so we do not place a significant emphasis on accuracy values. It should be noted that direct comparison of accuracy metrics from the minimized CAMEO test set against the baseline AF2 has a caveat: AF2 is not trained to predict minimized structures, whereas all our methods are. Similar to our results regarding structural quality, we find that finetuning on minimized data and with OpenMM-Loss has a larger increase in accuracy than utilizing either one of these approaches individually.

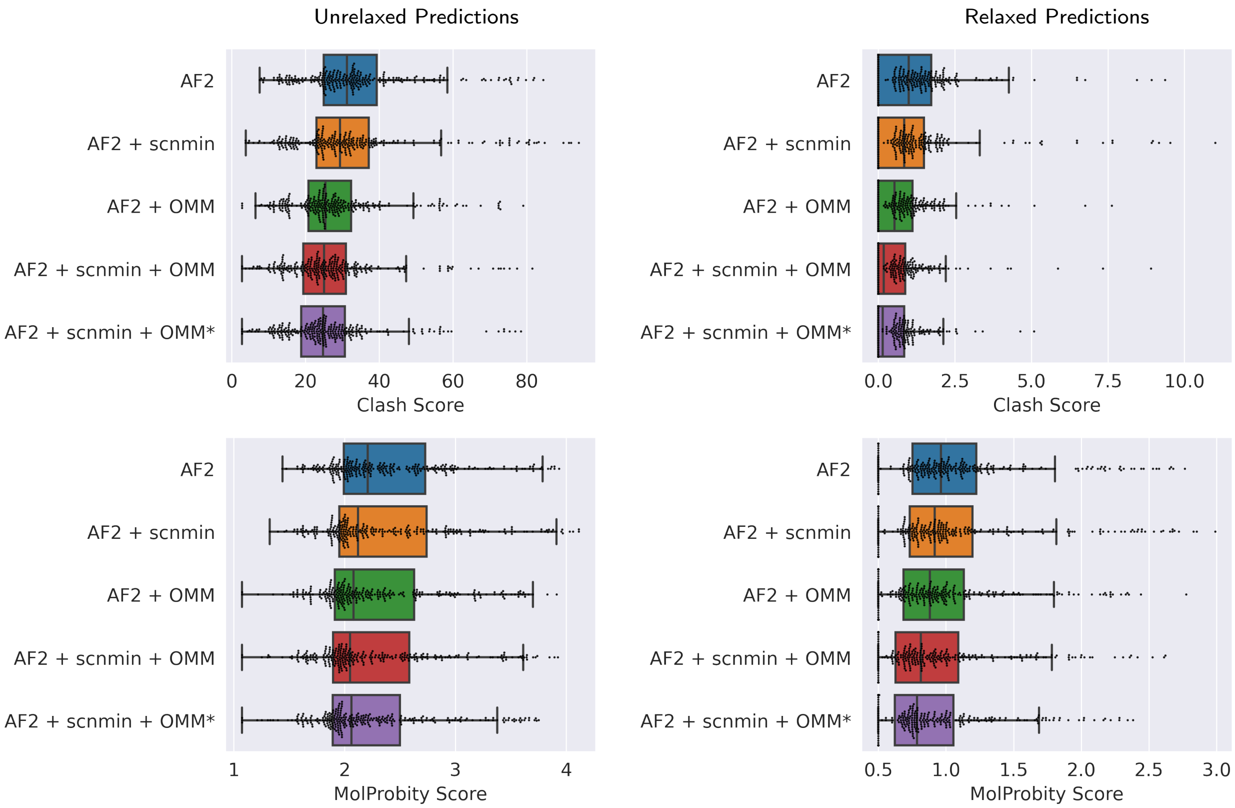


Figure 3.5: Evaluating the impact of the components of our method on structural quality metrics (Clash Score and MolProbity Score) relative to the **AF2** baseline. Each dot represents a single predicted protein from the CAMEO test set. Predictions are evaluated before and after relaxation (left and right columns, respectively). **AF2**: baseline AlphaFold2 initialized using `finetuning_5.pt`. + **scnmin**: finetuned on minimized SidechainNet proteins instead of unminimized structures. + **OMM**: finetuned using OpenMM-Loss with $\tilde{\sigma}$ activation. + **OMM***: finetuned using OpenMM-Loss with σ^* activation.

3.5 Discussion and Conclusions

If protein structure predictions are to become a valuable and practical tool for downstream tasks, the scientific community requires two things. First, we must improve accuracy at the sidechain level to avoid subtle changes in protein structure that have outsized impacts on tasks like molecular docking. Second, we require new paradigms to incorporate theoretical physical principles. By utilizing physical principles during training, it may be possible to continue

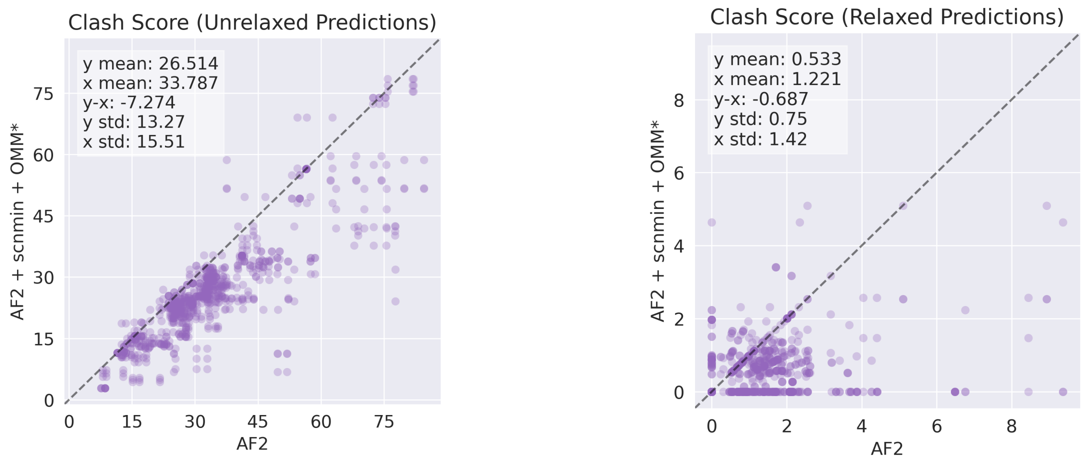


Figure 3.6: Comparing Clash Scores for predicted CAMEO test set proteins from one of our trained models (**AF2** + **scnmin** + **OMM***) versus the **AF2** baseline. Model predictions are evaluated before and after relaxation (left and right columns, respectively). **AF2**: baseline AlphaFold2 initialized using `finetuning_5.pt`. + **scnmin**: finetuned on minimized SidechainNet proteins instead of unminimized structures. + **OMM***: finetuned using OpenMM-Loss with σ^* activation.

	MolProbity Score (p-value)		**Clash Score (p-value)**	
Model	Unrelaxed	Relaxed	Unrelaxed	Relaxed
AF2	2.389	1.077	33.787	1.221
AF2 + scnmin	2.375 (4e-01)	1.087 (6e-01)	32.910 (3e-01)	1.186 (4e-01)
AF2 + OMM	2.285 (2e-02)	0.986 (1e-02)	28.156 (3e-06)	0.744 (3e-06)
AF2 + scnmin + OMM	2.271 (8e-03)	0.945 (8e-04)	27.143 (6e-08)	0.598 (2e-09)
AF2 + scnmin + OMM*	**2.244** (1e-03)	**0.915** (3e-05)	**26.514** (2e-09)	**0.533** (2e-12)

Table 3.1: Mean structure quality metrics of structure predictions before and after relaxation. Computed p-values reflect the significance of being lower than the AF2 baseline.

the steady march towards improved accuracy while enforcing that predicted structures, at a minimum, pass the basic test of physical plausibility.

Accuracy in this experiment is difficult to define because we have compared models trained with minimized and unminimized labels. Still, we provide a framework for improving other important metrics related to the structural quality of protein structure predictions. We also hope to instigate a more thorough discussion on the relationship between structural accuracy

Model	**lDDT$_{AA}$**	**lDDT$_{C\alpha}$**	**TM-Score**
AF2	0.736	0.818	0.671
AF2 + scnmin	0.744	0.826	0.679
AF2 + OMM	0.737	0.816	0.668
AF2 + scnmin + OMM	0.745	0.829	**0.682**
AF2 + scnmin + OMM*	**0.746**	**0.830**	**0.682**

Table 3.2: Structure prediction accuracy between unrelaxed predictions and the minimized CAMEO test set structures.

(e.g., lDDT) and quality (e.g., MolProbity) metrics which are not always congruent.

A major limiting factor of current protein structure prediction methods is the lack of ligand information at training or evaluation time. A loss function like ours could help models better discriminate between favorable and unfavorable conformations of proteins and protein-ligand complexes by evaluating the potential energy of predicted systems during training. A model capable of understanding even a modest level of chemical interaction principles may have a significant advantage over models agnostic to such principles. Though our framework only supports protein representations at present, future work may study the impact of using an energy-based loss function for models that include protein-ligand or ligand-only systems.

4.0 End-to-End Sidechain Modeling in AlphaFold2: Attention May or May Not Be All That You Need

This chapter is adapted from a manuscript in preparation for submission. J. E. K. developed the methods, conducted the experiments, and wrote the manuscript. D. R. K. provided guidance throughout.

4.1 Summary

AlphaFold2 has made significant strides in computational structural biology and drug discovery. However, limitations remain, particularly for downstream tasks such as molecular docking. We propose inaccuracies in amino acid sidechain prediction could contribute to these limitations. To address this, we explored three simple and complementary strategies: (1) substituting the default ResNet-based angle predictor in AlphaFold2 with a Transformer-like model, (2) integrating sequence-level convolutions into the angle predictor's encoder, and (3) refining the angle predictor using an energy-like loss function. Our analysis indicates that while ResNets and Transformers offer comparable performance, adding convolutions doesn't significantly enhance the predictor's performance. However, training with an energy-like loss can sometimes boost structural quality, especially when the entire model is finetuned. We suggest a holistic approach that looks beyond AF2's sidechain torsion angle predictor to improve sidechain modeling in future studies.

4.2 Introduction

4.2.1 The Significance of Sidechains in Drug Discovery

The discovery and design of novel drugs remain central to advancing medical science, with protein structures serving as critical components in structure-based drug discovery. While the unveiling of AlphaFold2[6] (AF2) marked a monumental leap in computational protein structure prediction, it's important to delineate its capabilities and limitations.

One significant limitation of AF2 lies in the use of its predictions for molecular docking, a technique vital for drug discovery[58–60]. Molecular docking often necessitates highly accurate predictions of sidechain orientations to forecast how potential drug molecules may interact with target proteins. Although AF2 offers impressive accuracy overall, its predictions often contain differences (some subtle, some not) that adversely affect the performance of subsequent docking methods. For instance, a study by He et al.[60] examined four small molecule-GPCR binding complexes and found that, in three of these complexes, the sidechain conformations predicted by AF2 deviated sufficiently from experimental structures, leading to altered docking results for known active compounds.

Furthermore, sidechains are central to the differences observed between *apo* (without ligand) and *holo* (with ligand) protein structures. Accurate sidechain modeling is essential as these differences often arise from shifts in sidechain torsional angles, which in turn affect configurational entropy and the local chemical environment[84,85]. These shifts in torsional angles are crucial to understanding the distinctions between *apo* and *holo* structures and are integral to the drug discovery process. Therefore, precise modeling of sidechains in both *apo* and *holo* states is imperative for developing effective drugs, underscoring the need for methods that can accurately predict sidechain conformations in various contexts.

4.2.2 The Evolution of Sidechain Modeling Techniques

Sidechain modeling, along with protein structure prediction in general, has undergone significant development in recent years. There are several problem formulations of note: sidechain packing, simultaneous modeling of protein sidechain and backbone atoms, and lastly, extensions of AF2 to improve its sidechain modeling capabilities.

The Sidechain Packing Challenge The sidechain packing problem has historically been one of the prominent challenges in protein structure prediction. The issue revolves around determining the optimal sidechain orientations on a fixed backbone. Several algorithms and heuristics have been developed over the years to address this challenge, with traditional approaches searching rotamer libraries for sidechain conformations with favorable energy[86,87]. Several machine learning approaches have also been developed[88–90], taking advantage of diverse deep learning model architectures to achieve state-of-the-art accuracy and speed.

Unified Backbone and Sidechain Prediction While the sidechain packing approach provides valuable insights, it simplifies the real-world scenario where sidechain and backbone conformations are intrinsically linked. Recognizing this interdependence, researchers have aimed to jointly predict both[6,32,91]. Methods that adopt this strategy often may also benefit from iterative refinement, where backbone and sidechain predictions inform and refine each other in a cyclic manner[6]. AlphaFold2 is a prime example of both unified protein backbone and sidechain prediction as well as iterative refinement. It is worth noting that AF2's sidechain predictions, though accurate, are still not as accurate as some of the contemporary sidechain packing techniques described above.

4.2.3 Potential Avenues for Improved Sidechain Modeling

Given AF2's impact, efforts to enhance its sidechain accuracy have also emerged. These include using traditional refinement methods using molecular dynamics (MD)[92] and docking pipelines[93], novel refinement methods[94,95], enhancing AF2 s input[96,97], or refinement of the AF2 model itself[98].

Building on these advancements, the integration of more sophisticated architectures and training techniques presents a potential avenue for improved sidechain modeling. For instance, considering the widespread success of the Transformer architecture across various domains, it stands as a promising replacement for the ResNet angle predictor in AF2. Adept at managing complex long-range dependencies, Transformers might be especially beneficial in modeling sidechain interactions and packing. In addition, research by Yang et al. (2023)[99] and prior work in our lab (Appendix A[100]) shed light on the utility of convolutional neural network layers in protein sequence and structure modeling. Yang et al. developed a convolution-only protein sequence model, while our earlier work developed an all-atom protein sequence-to-structure model that benefitted from convolution layers. These layers excel in discerning local spatial patterns and, when combined with attention mechanisms, might offer a comprehensive model that can capture localized and overarching interactions within proteins. Lastly, an alternative strategy is implementing an energy-like loss function to guide models based on the energetics of protein structures.

4.3 Methods

Data One of our underlying hypotheses is that protein structures with lower potential energy and fewer atomic clashes are more desirable for practical applications like structure-based drug discovery. Therefore, we utilize an energy-minimized subset (about 32,000 protein chains) of the CASP12 iteration of the SidechainNet dataset that was developed in prior work. The protein chains in this dataset were minimized using an energy-like loss function that interprets forces computed by OpenMM software as gradients. A validation set from CAMEO[80] was selected to match the set used in the OpenFold paper[70]. A test set of 93 proteins was selected from a one-year window of CAMEO proteins ending on January 3, 2023. Validation and test sets were minimized similarly to our training set. Structures that failed minimization were excluded from training and evaluation.

Training with Limited Compute Resources Training AlphaFold2 requires significant computational resources. To adapt to this challenge, we developed a pretraining method. This method is designed to utilize GPUs that are more readily available rather than the typically required A100s. We excised the angle predictor component from the original model to train independently of the full AF2 model. To do this, we performed inference for our dataset using the AF2 weights (`finetuning_5.pt` provided by the OpenFold authors[70]). During inference, we recorded the inputs to the angle predictor (the single sequence representation from AF2's Evoformer) and the target angle values from the true structure. We then trained each angle predictor model separately from the AF2 model on this data using the `supervised_chi` loss, which is a weighted combination of Mean Squared Error on predicted sin/cos angle values and on the magnitude of the sin/cos vectors to ensure they lie on the unit circle. After pretraining, angle predictors were reconstituted in the complete AF2 model, replacing the default ResNet. This enabled us to make progress while minimizing the use of computing resources.

We considered how best to evaluate the performance of our models relative to the pretrained OpenFold/AlphaFold models. The ResNet model trained as part of OpenFold is highly performant and has been trained for hundreds of thousands of training steps with an effective batch size of 128 on over 40 A100 GPUs and a massive amount of protein structure data. Since we do not have the compute budget to explore the impact of training all possible enhancements to AlphaFold2 for a similar amount of time, we evaluate each method from scratch and compare their performance after training for the same duration. We assume they may not perform as well as the baseline model, but our analysis will allow us to compare their relative performance.

4.4 Results

We investigate the impact of three modifications to AlphaFold2 to improve its sidechain modeling capabilities: (1) replacing AF2's ResNet-based angle prediction with a Transformer-based model, (2) supplementing the angle predictor with sequence convolutions, and (3) finetuning AF2's angle predictors to predict structures with lower energy.

4.4.1 Assessing Attributes of Attention

Because Transformer-like models for protein sequence modeling are highly performant, we posited that substituting them for AF2's ResNet could improve sidechain modeling. We executed a hyperparameter sweep over approximately 500 Transformer-like architectures and found several models comparable to the baseline ResNet from AF2. Models were first pretrained, as discussed above, entirely separately from the complete AF2 model.

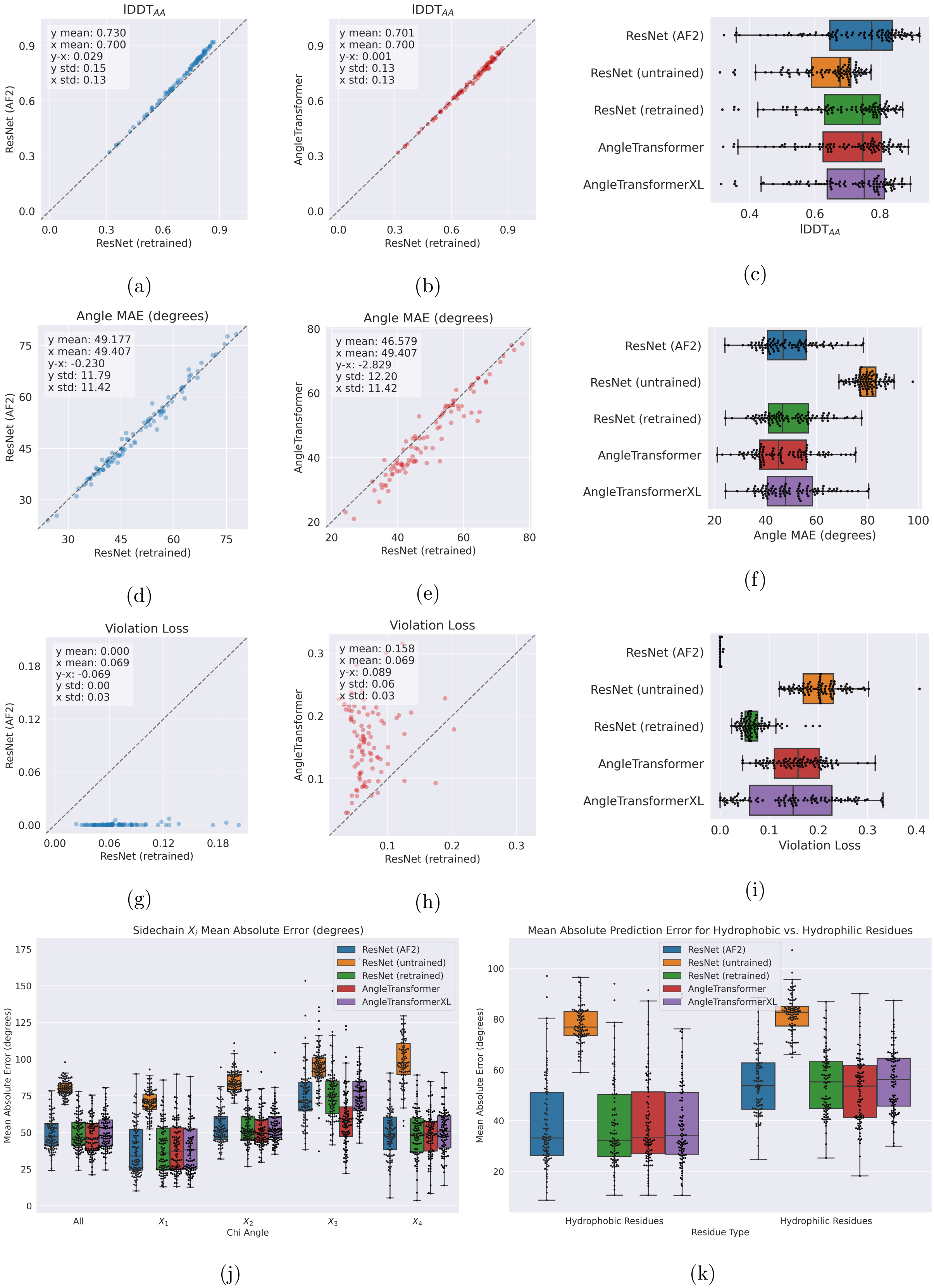
Comparison plots in Figures 4.1 and 4.2 evaluate the performance on various metrics for five models: (1) ResNet (AF2) - the pretrained ResNet directly loaded from the OpenFold model, (2) ResNet (untrained) - the ResNet architecture from AF2 without the pretrained weights, (3) ResNet (retrained) - where the untrained ResNet is trained from scratch, (4) AngleTransformer (AT)- the Transformer-like model from our hyperparameter sweep with the lowest Mean Absolute Error (MAE) on sidechain torsion angles from our validation set, and (5) AngleTransformerXL (ATXL), a AngleTransformer with a larger dimension architecture. ResNet (untrained) was included as a baseline to differentiate the impact of the angle predictor versus the rest of the AF2 model on structural accuracy. Model architectures are summarized in Table 4.1. Transformer models used GeLU activation.

Model	**Layers**	**Heads**	**Model Dim**	**Feed-Forward Dim**
ResNet (AF2)	2	-	128	-
ResNet (untrained)	2	-	128	-
ResNet (retrained)	2	-	128	-
AngleTransformer (AT)	42	1	64	1024
AngleTransformerXL (ATXL)	4	2	2048	256

Table 4.1: Comparison of ResNet and AngleTransformer model architectures.

Pretraining We first evaluated the models by pretraining them outside of the complete AF2 model, stopping training when the validation set loss angle MAE stopped improving. Results are seen in Figure 4.1. Despite the theoretical shortcomings of pretraining angle predictors without using the full model, we find that the ResNet (retrained) can recover a large portion of the all-atom structural accuracy (4.1a), measured by the All-Atom local Distance Difference Test (lDDT$_{AA}$). We also see that lDDT$_{AA}$ is mostly simlar across pretrained models (Figures 4.1b, 4.1c). Concerning sidechain torsion angle prediction, ResNet (retrained) can match the performance of the original ResNet (AF2) (Figure 4.1d), though it is outperformed by the AngleTransformer (Figures 4.1e, 4.1f). Figures 4.1j and 4.1k summarize the performance of each model across the four sidechain torsional angles and by amino acid hydrophobicity. Lastly, in Figures 4.1g, 4.1h, and 4.1i, we see that our pretraining procedure resulted in predictions with higher violation loss that the fully trained ResNet (AF2) model.

Pretraining and Finetuning We hypothesized that we would get a complete picture of the AngleTransformer and ResNet's strengths if we finetuned each further, reconstituting them into the full AF2 model and following standard finetuning procedures. The results of



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 4.1: **Pretraining:** Comparing AF2 angle predictor structure accuracy on held-out CAMEO test set proteins after pretraining only. Higher lDDT$_{AA}$ is better. For violation loss and angle MAE, lower is better.

this finetuning experiment are summarized in 4.2 and Table 4.2. The AngleTransformerXL was excluded from finetuning due to limited computational resources.

After finetuning the ResNet and AngleTransformer, we find that both increase in accuracy with respect to MAE and lDDT$_{AA}$, with the AngleTransformer outperforming the ResNet (retrained) on MAE by a few degrees (Figure4.2e and Table 4.2, p=0.05) and the ResNet insignificantly more performant according to lDDT$_{AA}$ (Figure4.2b and Table 4.2, p=0.27). Violation loss values have improved for both models after finetuning, with ResNets still having an advantage (Figures 4.2h, 4.2i). The AngleTransformer is slightly better at predicting hydrophilic residue torsion angles than the ResNet (Figure4.2k).

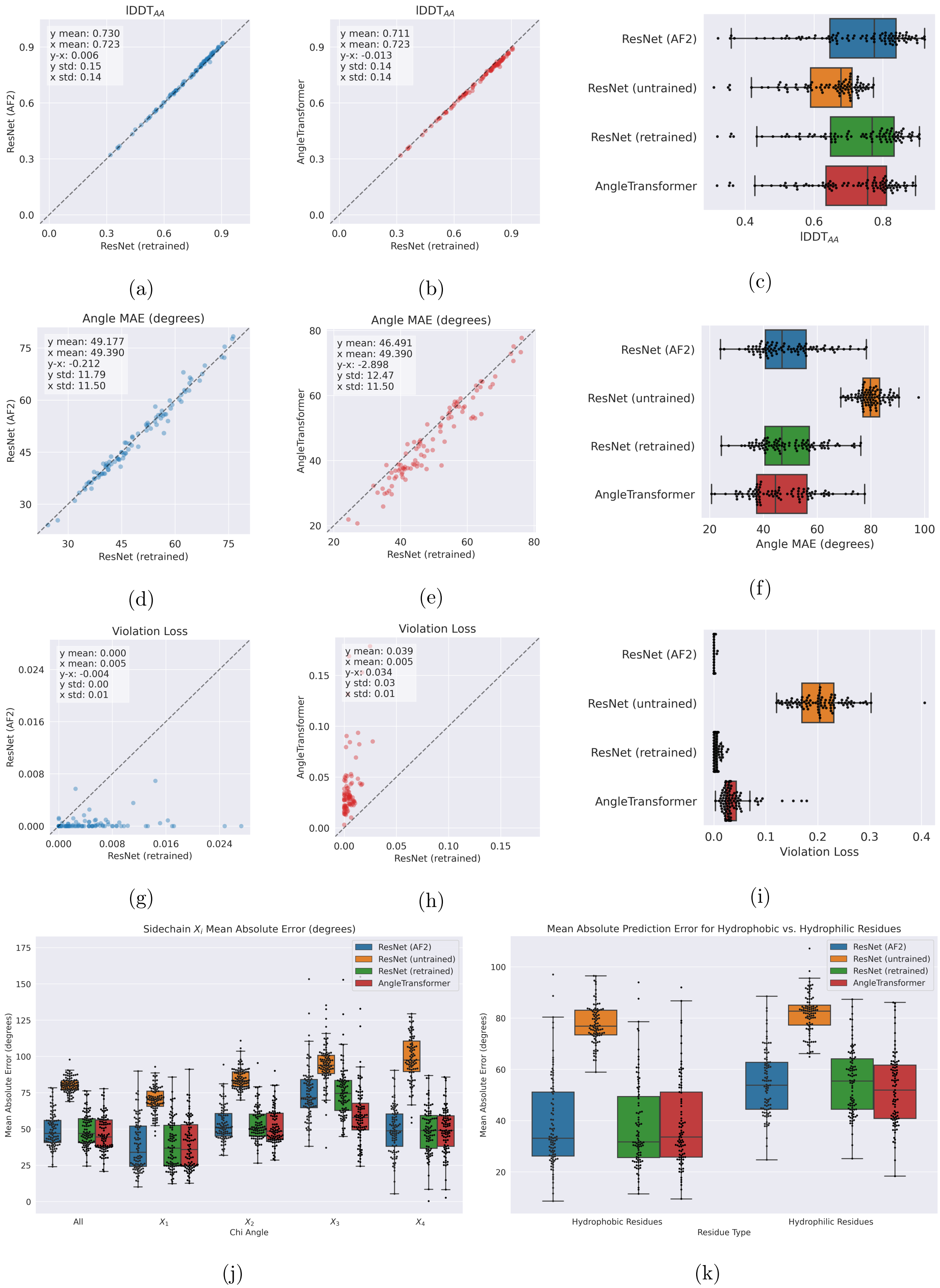
We also compare the MAE of each model across amino acid residue identity in Figure 4.3.

Model	**lDDT$_{AA}$**	**MAE (degrees)**	**Violation Loss**
ResNet (AF2)	**0.730**	49.177	**0.0004**
ResNet (untrained)	0.642	80.208	0.2051
ResNet (retrained)	0.723	49.390	0.0048
AngleTransformer	0.710	**46.491**	0.0389

Table 4.2: Accuracy on CAMEO test set structures after pretraining and finetuning procedures.

4.4.2 Capturing Convolutional Characteristics

Prior work (See Appendix A)[100] has shown that for all-atom protein structure modeling with Transformers, adding convolution layers before the Transformer encoder typically enhances prediction on angle prediction tasks. We applied the best-performing convolution architecture from our prior work in Appendix A, a one-dimensional convolution in the sequence dimension with a kernel size of 11 that preserves the sequence length and the dimensionality of the sequence representation. We compare how adding this convolutional layer impacts the performance of both our ResNet and AngleTransformer models in Figure 4.4. We repeated



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 4.2: **Pretraining and Finetuning:** Comparing AF2 angle predictor structure accuracy after finetuning models from Figure 4.1. Higher lDDT$_{AA}$ is better. For violation loss and angle MAE, lower is better.

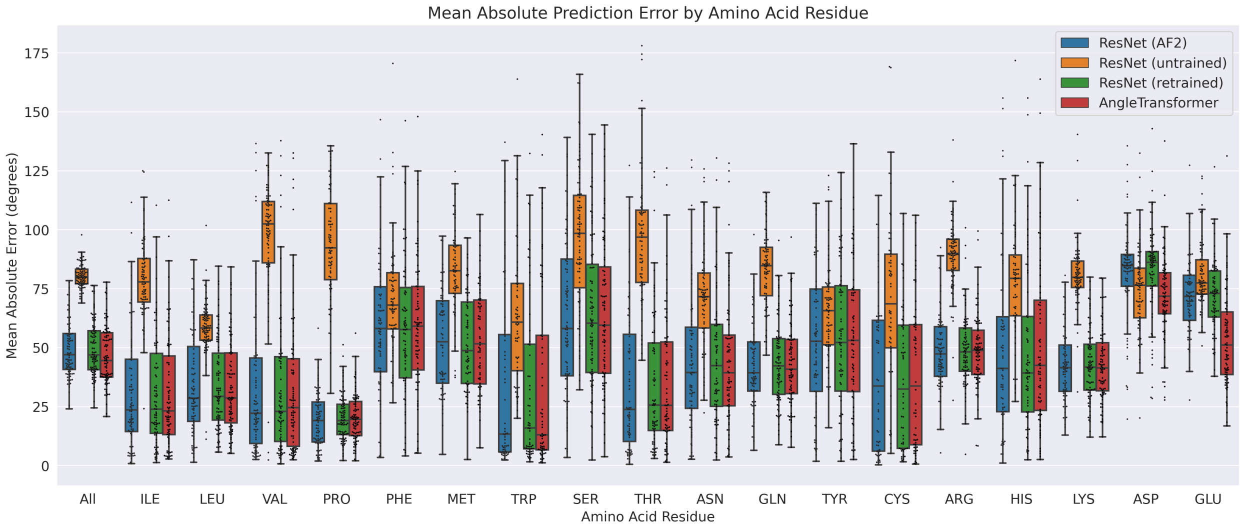


Figure 4.3: Comparing AF2 angle predictor angle accuracy by residue identity. Lower Mean Absolute Error is better.

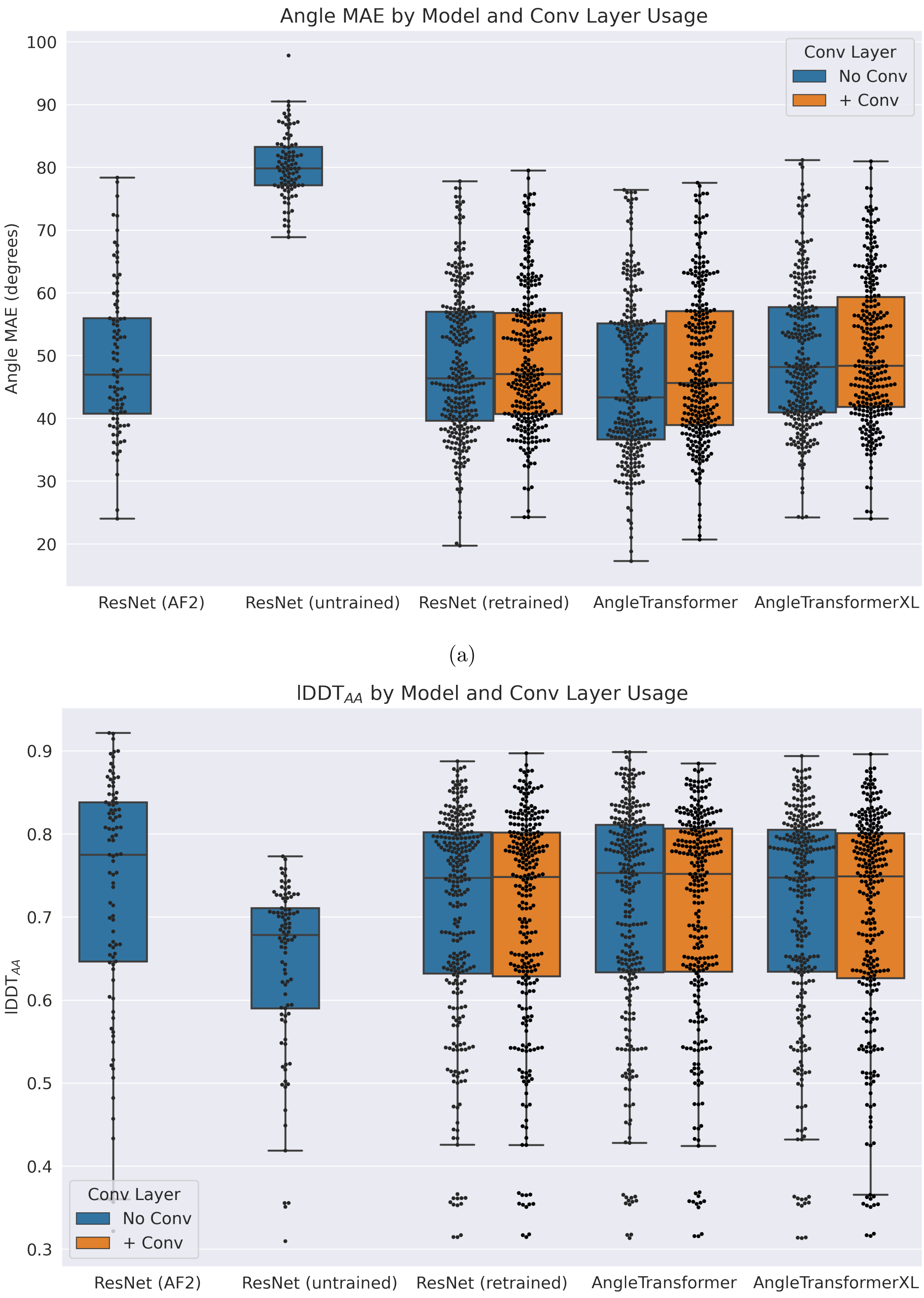
the experiment with three initial random seeds for each model tested. All predictions of each model type were combined before plotting.

Despite prior evidence suggesting that convolutional layers may supplement attention-based models[100] or remove the need for them entirely[99], adding a convolutional layer at the start of the ResNet and AngleTransformer did not significantly alter angle predictor performance on MAE (Figure 4.4a) or lDDT$_{AA}$ (Figure 4.4b).

4.4.3 Evaluating Energetic Elements

Lastly, we utilize an energy-based training procedure called OpenMM-Loss to improve the structural quality of predicted structures. The process is described in detail in Aim 2 of this dissertation.

To find the experimental conditions resulting in the highest accuracy, we evaluated several different variables in tandem: using the ResNet vs. AngleTransformer, OpenMM-Loss (OMM) vs. standard finetuning (no-OMM), and various finetuning procedures that finetuned only the angle predictor or the entire AF2 model.



(a)

(b)

Figure 4.4: Evaluting the effect of sequence convolutions during pretraining. Lower MAE and higher lDDT$_{AA}$ are better.

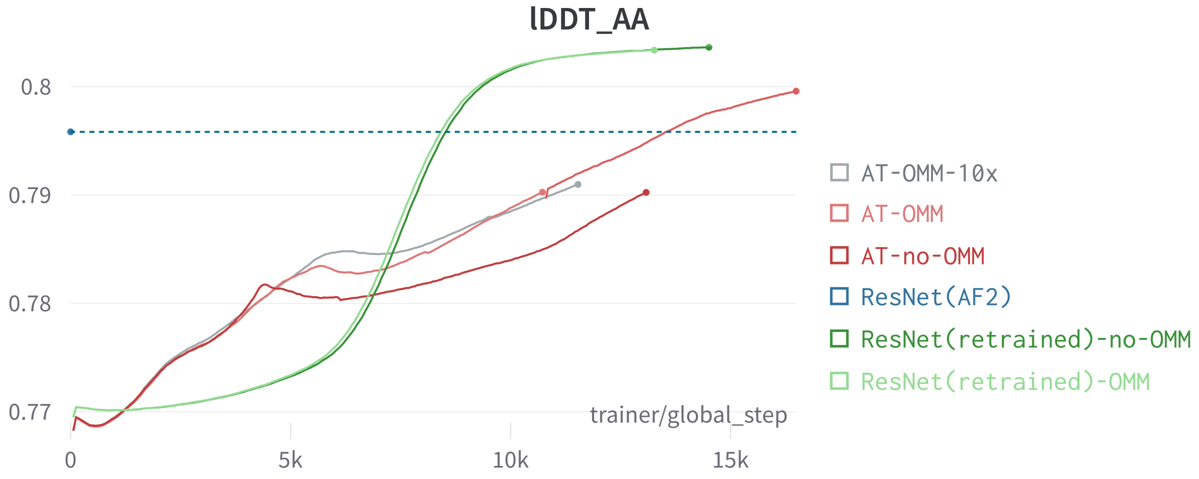


Figure 4.5: Evaluting the effect of OpenMM-Loss on training angle predictors. Higher lDDT$_{AA}$ is better.

Finetuning Angle Predictors With and Without OpenMM-Loss In our first set of experiments, we took the ResNet (retrained), hereafter referred to as "ResNet," and AngleTransformer models developed above and reconstituted them into the complete AF2 model. We finetuned them as in Section 4.4.1 while testing the effects of training with and without OMM. In one experiment, AT-OMM-10x, we also increased the weight of the OpenMM-Loss from 0.01 to 0.1. All models were trained for about one week using four A100 GPUs. ResNets and AngleTransformers significantly differed in their training behavior (Figure 4.5). ResNets, in general, benefitted from much faster training times and reached their maximal accuracy values in the shortest number of training steps. In the allotted compute budget, it is unclear if the AngleTransformer reached its maximum accuracy. Accuracy and structural quality metrics for four models (AT-OMM, AT-no-OMM, ResNet-OMM, and ResNet-no-OMM) are summarized in Figures 4.6, 4.7, and 4.8. The effects of increased OMM weight for the AT-OMM-10x model are summarized in Figure 4.9. Note that that the AngleTransformer used here includes a convolution layer because at the start of this experiment, we believed it would be the most performant.

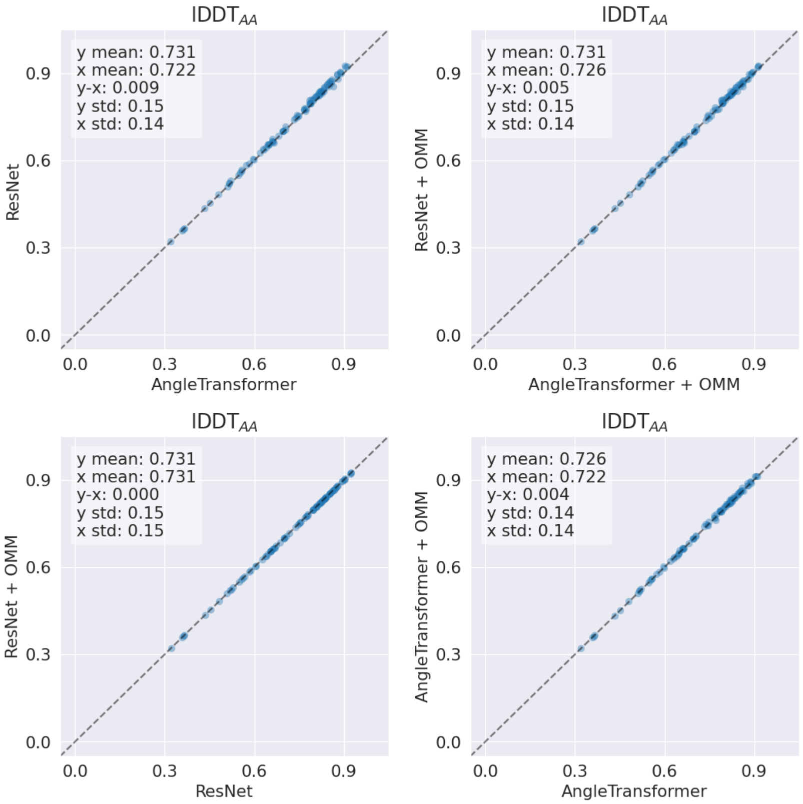


Figure 4.6: Comparing lDDT$_{AA}$ after finetuning full AF2 models across two variables: AngleTransformer vs ResNet, and OpenMM-Loss (OMM) vs standard AF2 loss. Higher lDDT$_{AA}$ is better.

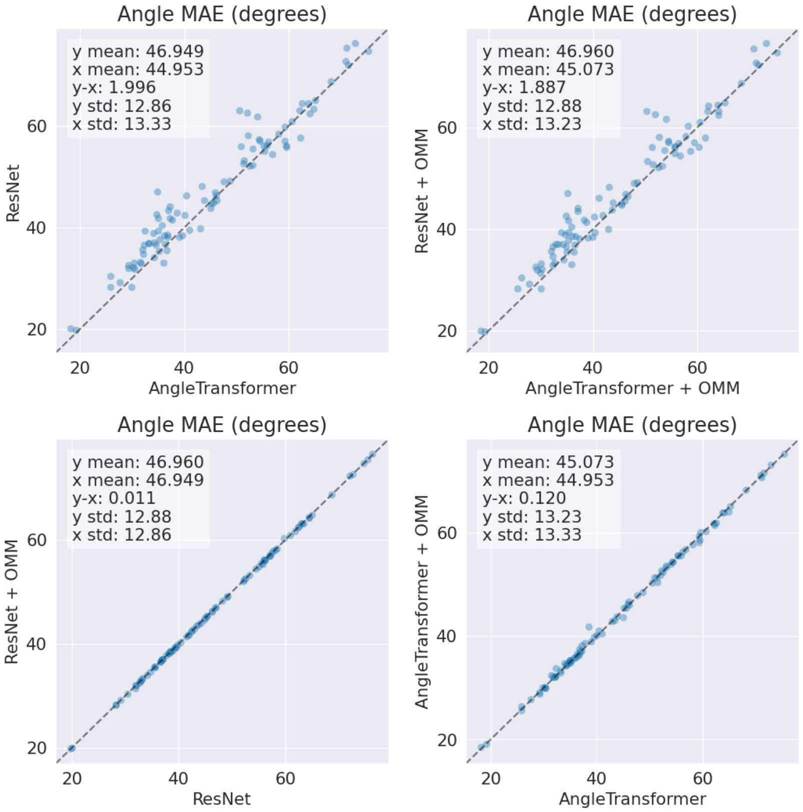


Figure 4.7: Comparing angle MAE after finetuning full AF2 models across two variables: AngleTransformer vs ResNet, and OpenMM-Loss (OMM) vs standard AF2 loss. For angle MAE, lower is better.

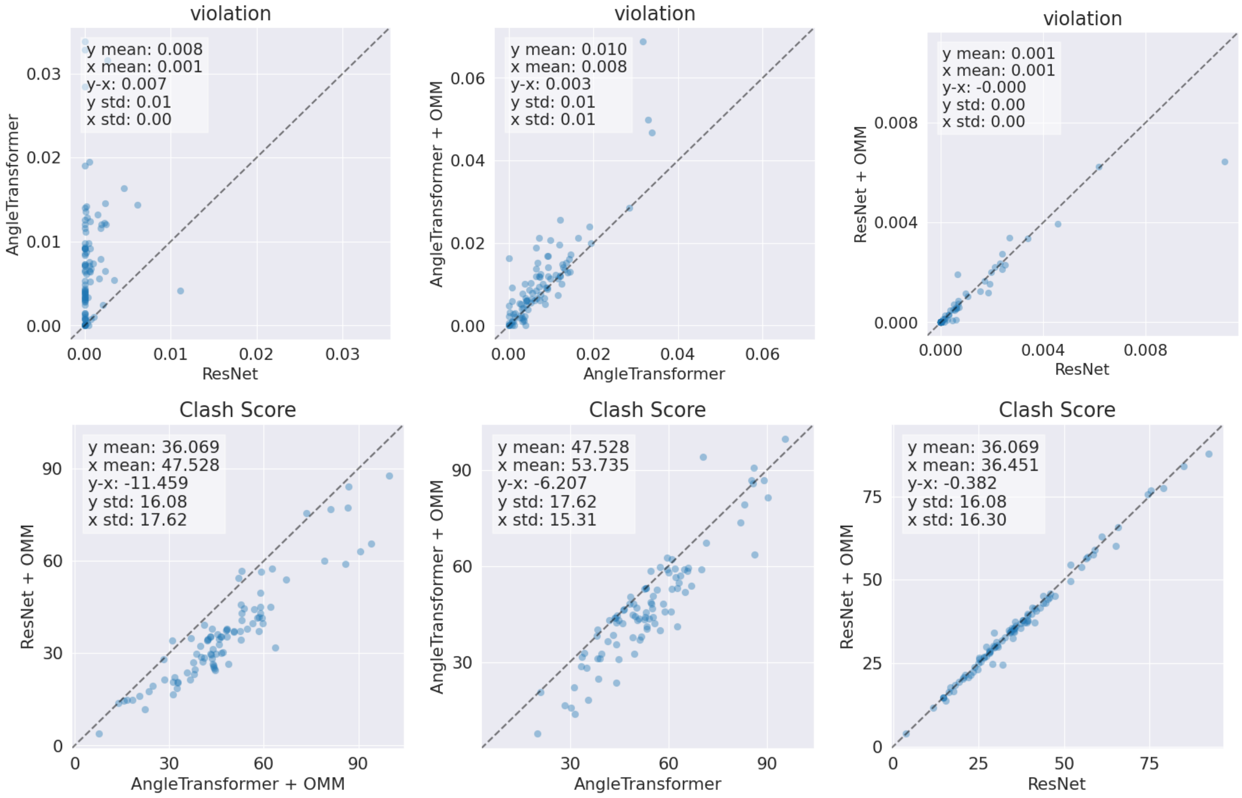


Figure 4.8: Comparing structural quality metrics after finetuning full AF2 models across two variables: AngleTransformer vs ResNet, and OpenMM-Loss (OMM) vs standard AF2 loss. For violation and clash score values, lower is better.

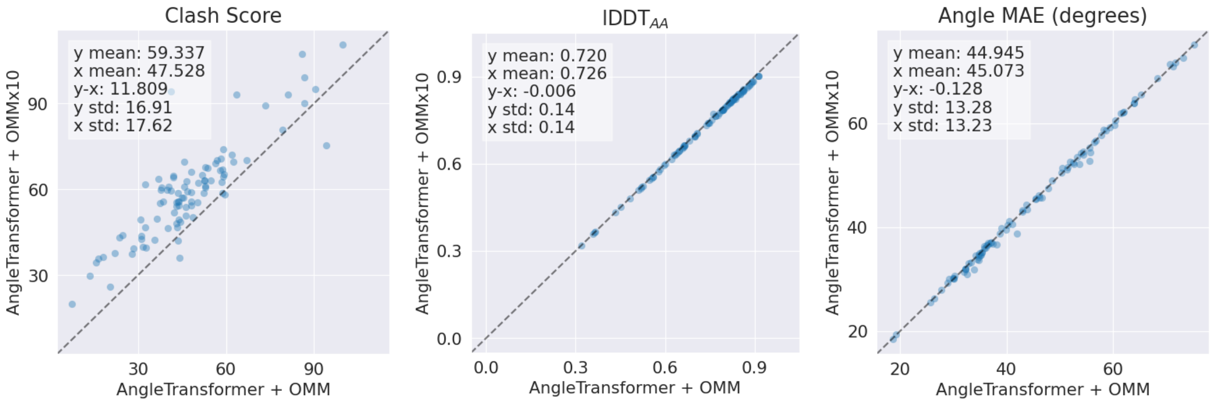


Figure 4.9: Evaluating the impact of increased OMM loss weight (Angletransformer + OMMx10) on structural quality and accuracy. Higher lDDT$_{AA}$ is better. For clash score and angle MAE, lower is better.

FrankenModels: Combining Previously Finetuned AF2 Models with New Angle Predictors We hypothesized that better performance could be achieved by training the angle predictors (ResNets and AngleTransformers) in conjunction with the rest of the AF2 model rather than only finetuning the angle predictor weights as in the prior section. However, this requires additional computational resources. As an alternative, we proposed taking the pretrained models from Section 4.4.1 and combining them with complete AF2 models that had been finetuned in Aim 2. Though the angle predictors and AF2 base model had been finetuned on our SidechainNet minimized dataset independently, their combination may limit the required additional finetuning time since the models from Aim 2 have already been trained for several weeks. We refer to these AngleTransformer models as "AT-FrankenModels" due to swapping out different angle predictor "heads" for AF2. Training curves displaying the performance of AT-FrankenModels can be seen in Figure 4.10. Plots in Figure 4.11 summarize their performance relative to the AngleTransformer models. Both AT-FrankenModels initially underperformed during training but quickly surpassed the performance of the other models.

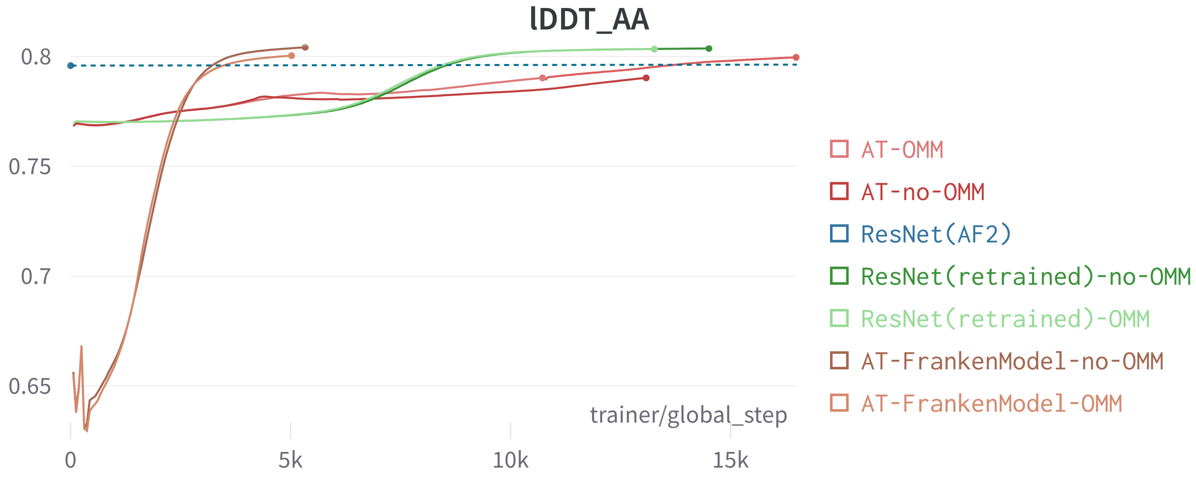


Figure 4.10: Combining pretrained whole AF2 models with our separately pretrained AngleTransformer into the AT-FrankenModel. Higher lDDT$_{AA}$ is better.

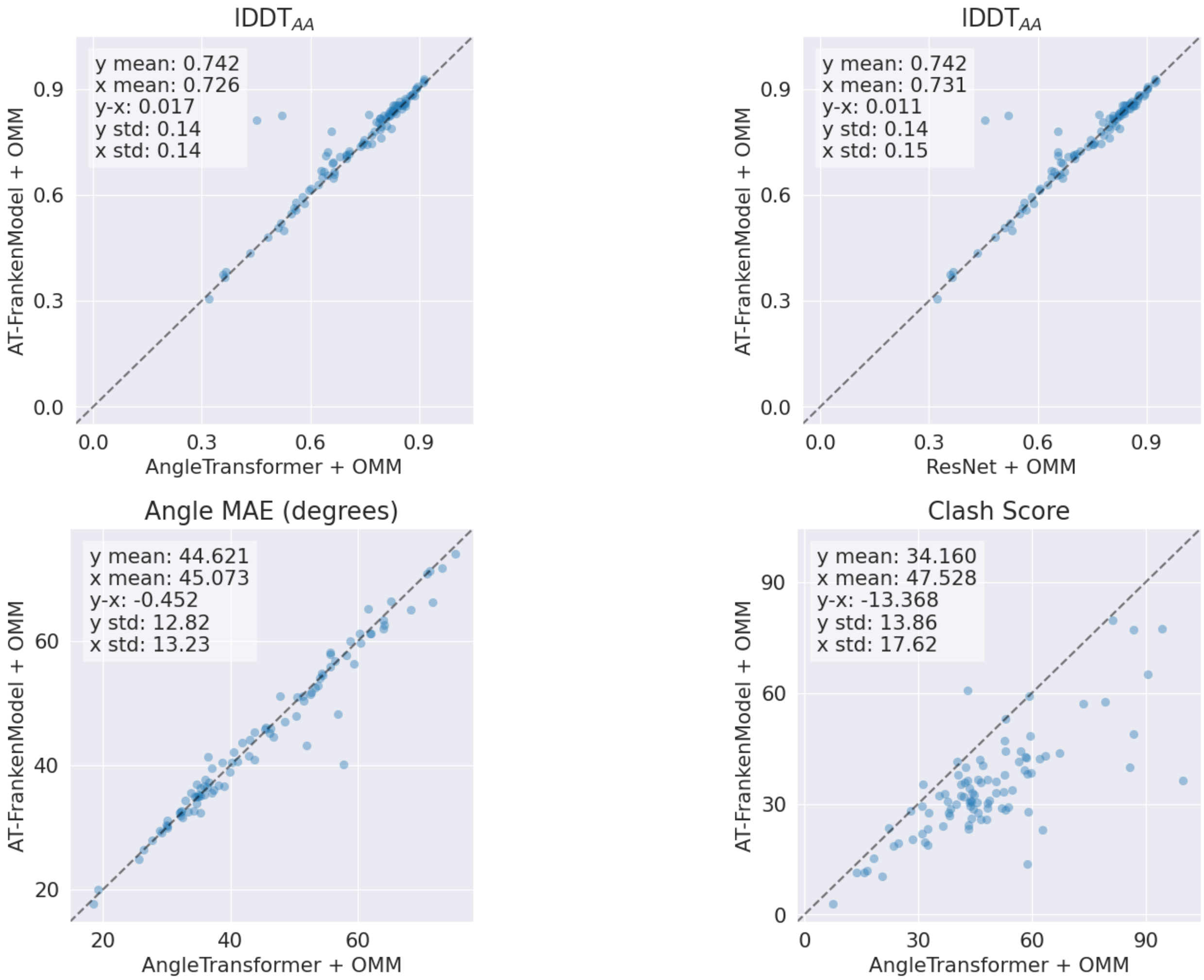


Figure 4.11: Evaluating the accuracy of "AT-FrankenModels" finetuned with OMM Loss on the CAMEO Test set. AT-FrankenModels have angle predictors and base-AF2 model components that were pretrained separately before being finetuned together. Higher lDDT$_{AA}$ is better. For clash score and angle MAE, lower is better. If you read this far, email jonathanking.jek@gmail.com, and I will personally thank you for spending your free time reading this voluminous tome.

4.5 Summary

We explored Transformer-like architectures as alternatives to ResNets for predicting sidechain torsional angles in AlphaFold2 (AF2). The AngleTransformer consistently surpassed ResNets in sidechain angle accuracy (measured as MAE), with a smaller model (AngleTransformer) outperforming a larger one (AngleTransformerXL). Improved MAE was observed for distal sidechain chi angles (χ_{2-4}), especially for flexible residues like glutamic and aspartic acids (Figure 4.3). However, the ResNet models surprisingly excelled in violation loss, even though they were only pretrained to minimize angle loss (Figure 4.1).

Despite our intuition, adding convolution layers to AF2's angle predictors did not enhance accuracy (Figure 4.4). Furthermore, in experiments combining AngleTransformers, OpenMM-Loss, and various training strategies, ResNets proved more efficient, achieving maximum accuracy quicker (Figure 4.5). Eventually, both model types showed similar structural accuracy, with AngleTransformers displaying lower angle MAE (Figure 4.7).

Training with OpenMM Loss didn't significantly alter accuracy (Figures 4.6 and 4.7), and increasing its weight didn't yield improvement (Figure 4.9). While OpenMM Loss affected the training versus accuracy dynamic, its implementation didn't consistently reduce violation loss values (Figure 4.8). Still, models trained with OpenMM Loss had lower clash scores than models trained without. That OpenMM Loss reduced the number of clashes in the AngleTransformer but not the ResNet points to the AngleTransformer's capacity to improve metrics other than angle MAE.

Upon shifting focus from angle predictors, we introduced AT-FrankenModels, combining pretrained AngleTransformers and AF2 models to reduce the computational cost of full-AF2 finetuning. Initially, AT-FrankenModels underperformed but swiftly improved (Figure 4.10), eventually surpassing AngleTransformer and ResNet predictors, which have not had the benefit of extended pretraining (Figure 4.11).

4.6 Discussion and Future Work

We observed promising options for improving protein structure prediction. Although our AngleTransformer models predicted sidechain torsional angles with lower MAE, this metric alone doesn't fully address the complexity of improving sidechain modeling in AF2.

Our findings suggest that while AngleTransformers may offer more accurate modeling, evidenced by lower angle MAE and fewer clashes when trained with OpenMM Loss, the advantages are not clear-cut. In contrast, ResNet models provided equal or better lDDT_{AA} scores, lower violation loss, required less training time, and were more parameter-efficient.

Despite previous studies suggesting the effectiveness of convolution layers, we found limited utility for them in this context. The benefits of convolutions, such as recognizing local structures and patterns, might be overshadowed by AF2's existing attention mechanisms and the specialized feed-forward layers in its ResNet angle predictor. Performance might improve with more extensive hyperparameter tuning for convolutional layers.

Our experiments underscore the importance of a comprehensive approach to improving sidechain-level protein structures. While refining AF2's angle predictor can enhance sidechain modeling, finetuning the entire model yields the best accuracy and structural quality results. Future work could explore alternative sidechain prediction representations, possibly eschewing χ-angle prediction for atomistic modeling of sidechain-backbone interactions. Another worthwhile avenue might be implementing more efficient attention mechanisms, balancing improvement in expressiveness with cost-effectiveness. Alternatively, attention mechanisms might be supplemented by including a geometric prior, constructing attention weights based on distances or other geometric feature vectors measured from the data.

5.0 Conclusions and Future Directions

5.1 Conclusions

5.1.1 Evolution of Protein Structure Prediction

Protein structure prediction remains a pivotal aspect of computational biology, even with impressive advancements like AF2. The inception of AF2 marks a synergistic blend of decades of machine learning methodology development and breakthroughs in quantitative biology and data. The field has mirrored the progressive trajectories observed in Natural Language Processing (NLP) and Computer Vision (CV), where competitions have consistently driven innovation and excellence. In a similar vein, contests like CASP and CAMEO continue to energize the domain of protein structure prediction, offering platforms for methodological comparison and benchmarking that evolve with the field.

5.1.2 Current Landscore

AF2 has undoubtedly made its mark on computational biology. Along with the publication of its underlying methodology and code, the authors have also released more than 200 million predicted protein structures as part of the AlphaFold Structure Prediction Database, all of which have garnered widespread attention. While many of these predictions are useful, a notable number have serious issues. Nonetheless, for a range of applications, the current predictions are often adequate, and we anticipate continuous enhancements to AF2 and related methodologies in the near future.

5.1.3 Contributions of This Work

This dissertation presents three distinct, interrelated pathways aimed at advancing protein structure prediction, conceptualized and developed in both the pre and post-AF2 eras:

1. **SidechainNet:** Introduced to facilitate access to and engagement with the field, SidechainNet provides a user-friendly dataset accompanied by preprocessed data and helper functions. It has been embraced by researchers from various backgrounds, serving as a practical entry point and valuable resource for the community.

2. **OpenMM Loss Toolkit:** This toolkit helps models like AF2 better understand biophysics. While AF2 provides accurate predictions, it doesn't offer insights into how proteins fold. The OpenMM Loss Toolkit works to address this gap, mitigating clashes and structural irregularities in AF2's predictions, with further testing of its applications underway.

3. **AF2 Sidechain Modeling Enhancements:** Our experiments aimed to enhance AF2's sidechain modeling capabilities. While no model clearly outperformed AF2, the experiments provided insights into the potential strengths and limitations of different models. Transformers, provided adequate data and computational resources are available, could potentially increase accuracy, although the additional computational complexity is a consideration. (ResNets reach an lDDT$_{AA}$ of 0.80 in about $^1/_2$ of the number of optimization steps as AngleTransformers, all other training parameters being the same).

5.1.4 Looking Ahead: Future Directions

There's more work to be done in protein structure prediction.

- **Understanding Poor Docking Outcomes:** High-accuracy predictions do not invariably translate to successful docking outcomes, as evidenced by the work from Masha et al.[58] and others. The precise reasons for these discrepancies are unclear, and there's no definitive roadmap for rectifying these in AF2.

- **Protein Dynamics:** Proteins, in reality, are dynamic entities, not static structures. The field must evolve its methodologies to capture the full spectrum of protein motions, from minor atomic jostles to substantial domain movements that are integral to enzymatic activities.

- **Emerging Architectures:** Generative models like diffusion models are gaining traction. These models allow for sampling from the data distribution on which they are trained, holding potential for innovative applications such as generating libraries of potential ligands and their conformations relative to protein structures.

- **Beyond MSAs:** New methods need to be devised to circumvent AF2's reliance on multiple sequence alignment (MSA) data. While attempts like ESM-Fold[56], which leveraged a language model, showed promise, they fell short of AF2's performance, indicating that further development is needed in this area.

- **Energy-Based Training Procedures:** Our exploration into energy-based training methodologies aims to inspire the development of machine learning techniques that are not opaque "black boxes". By integrating understood chemical and physical principles with advanced machine learning techniques, we envision creating a framework that enables more informed and accurate predictions and decision-making in the field.

In conclusion, this dissertation contributes practical tools and insights to the domain of protein structure prediction, laying a solid foundation for future investigations and developments. We are determined that combining basic science with advanced computing will play a pivotal role in developing better predictive models.

Appendix A: Evaluating the Impact of Sequence Convolutions and Embeddings on Protein Structure Prediction

The following report describes an original method for all-atom protein structure predictions using Transformer models developed by J. E. K. It was written and published by J. E. K. on the Weights and Biases platform on November 3, 2022. Weights and Biases is a set of cloud-based machine learning tools, as well as a community for machine learning developers. Thanks to Nicholas Bardy, who implemented molecular visualizations for this project.

At the time of writing, an interactive version of this report is available at the following address: `https://wandb.ai/koes-group/protein-transformer/reports/Evaluating-the-Impact-of-Sequence-Convolutions-Embeddings-on-Protein-Structure-Prediction--Vmlldzo2OTg4Nw`

A.1 Introduction

Despite recent advancements in the fields of biology, computer science, and machine learning, protein structure prediction remains one of the "holy grails" of molecular biology. Since the Nobel Prize-winning discovery of the structure of sperm whale myoglobin in the late 1950s, researchers have worked to uncover the 3D structure of thousands of other proteins in an effort to better understand the molecular basis of life.

A.1.1 Protein Structure Background

Proteins, long chains of amino acids that naturally fold up into unique, “globular” structures when produced, are essential to many of the life-sustaining chemical reactions in the cell. Even though proteins are composed of only about 20 different types of amino acid residues, each residue has a unique molecular “side chain” with chemical properties that determine the protein’s function. In fact, knowing the precise orientation of each amino acid is key to structure-based drug discovery, a method of developing new medicines by rational examination of where a drug molecule might bind to its protein target.

Although protein structure information is undeniably significant, it is difficult and expensive to produce. For instance, scientists may spend upwards of tens of thousands of dollars to produce a protein structure via X-ray crystallography, with many failed experiments along the way. As a result of the high cost, high impact, and low availability of protein structures, the scientific community has spent significant time and effort on methods that can predict the shape of proteins from their primary amino acid sequence, since this information is much more readily available.

To put this in perspective, as of March 2020, there were more than 175 million protein sequences available in the UniProtKB database, while only 162 thousand protein structures in the Protein Data Bank (PDB) The availability of protein structure data (red) pales in comparison with that of protein sequence data (blue) in Figure A.1. **Clearly, there is a great need (and opportunity) to predict protein structures from their underlying amino acid sequences!**

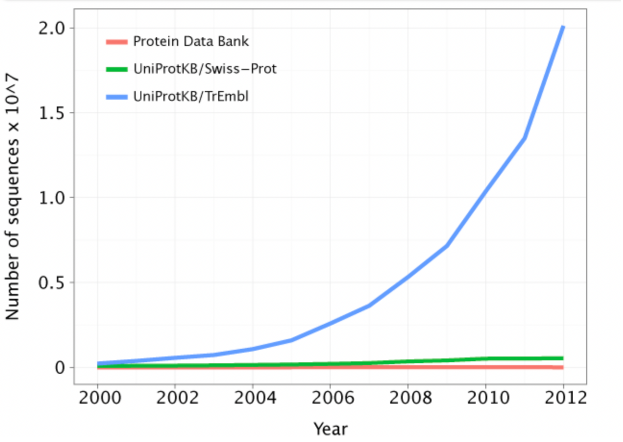


Figure A.1: The growth of protein sequence vs. structure data. Figure from GORBI[101], used with permission.

A.2 My Work

Since August 2017, I have been working on deep learning methods for protein structure prediction. My latest attempt is to harness the power of the "Neural Machine Translation" methods that have been so successful over the last few years. If I can formulate the problem of protein structure prediction as one of language translation, then I can take advantage of these high-performing models!

In my case, I am translating from the "language of amino acid residues" (lysine, arginine, etc.) into the "language of angles" that define how each atom is placed with respect to its predecessors. The angles are then converted into Cartesian coordinates, which can be directly compared to the true protein structure. One of the main contributions of my work is the fact my models will predict both the protein backbone and side chain atoms, which is imperative for certain research like structure-based drug discovery.

The model I am using is based on the now-ubiquitous Transformer model[14], although my current model disposes of the decoder half for simplicity. The training data is based on ProteinNet[43] by Mohammed AlQuraishi, but has been modified to include sidechain information.

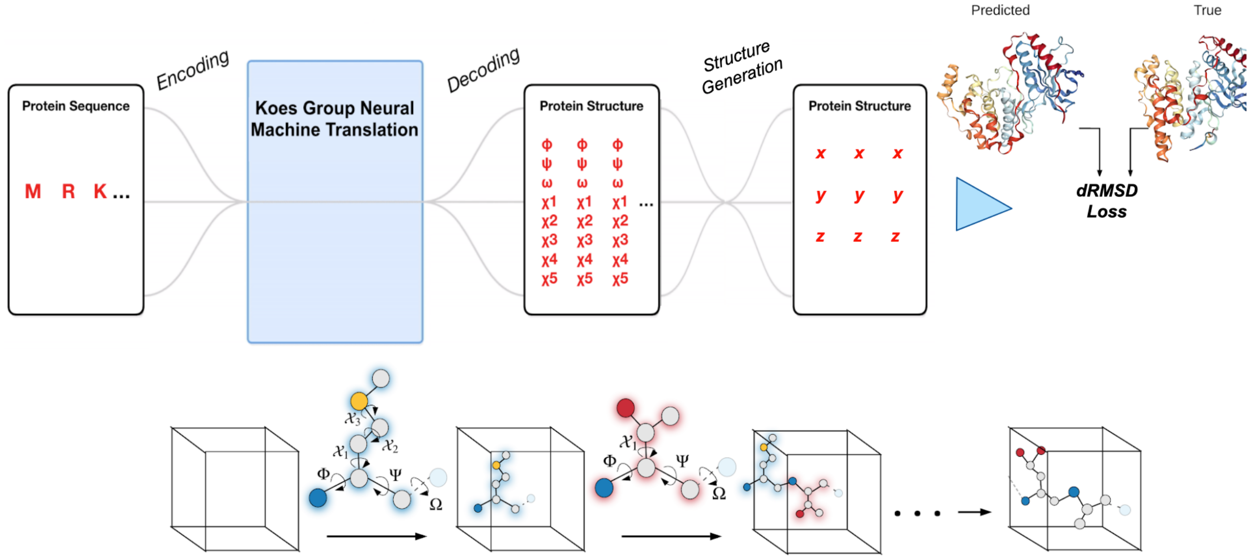


Figure A.2: The ProteinTransformer method architecture.

You can find my current work-in-progress on the ProteinTransformer (Figure A.2) here.

A.3 Models Trained with MSE Loss Improve by Adding Convolution Layers

A.3.1 Motivation

As much as we've tried, we've been having some trouble getting our ProteinTransformer to make reasonable predictions. Many proteins predicted so far kind of look like big balls of spaghetti! It could be the case that since our training data is based on amino acid sequences alone rather than incorporating information common to other prediction methods (think multiple sequence alignments, etc.), there just may not be much signal to work with.

However, another hypothesis is that while Transformer layers are great for predicting long-range interactions between amino acid residues, they may not incorporate local interactions as well. Adding local representations of the sequence via 1D convolutions may be enough to help the model learn simple things, such as the location of alpha-helices, that are only dependent on their nearby neighbors.

A.3.2 Experiment

In this experiment, I modified my base ProteinTransformer model by adding 1-dimensional sequence convolution layers after the embedding layer, but before the Transformer layers. In essence, the predictions from this model look something like this:

$$\text{Pred} = \text{TransformerAttention}_{1..L}(\text{Conv}_{1..n}(\text{Embedding}(X)))$$

I tested two kernel sizes, 3 and 11. I also experimented with the number of output channels from each convolution layer. In the models titled `conv-enc-3/2` and `conv-enc-11/2`, each convolution layer had twice as many output layers as input layers. This is common practice in convolutional neural networks for image processing as each layer builds up a representation of the underlying data. Models ending with `3/1` or `11/1` had the same number of input and out channels. See Figure A.3 for training results.

Each model was trained for 10 epochs with the same hyperparameters and using the Mean Squared Error (MSE) loss between the predicted angles and the true angles that represent the protein's structure in 3D space. For reference, lower loss values mean better performance.

This is good news - our hypothesis was correct! It looks like convolution layers (all lines except blue, Figure A.3) help the model perform slightly better, especially when the convolution layer windows are slightly larger (yellow, orange). Not all of the predictions look that great, but the point of this experiment is to find any improvement, and for right now, I'm satisfied!

A.4 Models Trained with DRMSD Loss Also Improve with Convolution Layers

A.4.1 Motivation

Protein structures are more than just angles, though. From Mohammed AlQuraishi's work, we know that "Distance-Based Root Mean Squared Distance", or DRMSD, is one differentiable way to compare two protein structures and train a model.

Here, I am repeating the same experiment as above, but instead of just training the models on the RMSE between the true and predicted angles, I am also comparing the complete protein structures in something called the "Combined Loss" which combines both the RMSE and DRMSD values. Training results are summarized in Figure A.4.

Great! We've now verified that convolution layers are helpful, regardless if we are optimizing for RMSE loss or a combined loss.

A.5 Embedding Layers Appear to Improve Overall Performance

A.5.1 Motivation

Okay, now we know that adding convolution layers to the mix can help our model overall! However, we started wondering whether or not the embedding layer was really necessary for our model.

You see, in language translation and other language processing tasks, embedding layers are used to take a high-dimensional representation of a word ($d \approx 10^4$) and turn it into a lower-dimensional representation ($d \approx 10^3$) that incorporates the "meaning" of the word.

However, we are not dealing with a very large input vocabulary! In fact, since there are only 20 amino acids, we have $d = 20$. So, what's the point of the embedding? Well, maybe our embedding layer is still learning something important about each residue and incorporating this information into its embedding during training. Let's see!

A.5.2 Experiment

The following runs (Figure A.5) show many examples of different convolution layer patterns that are repeated with and without embedding layers. These models are trained such that the last convolution layer has the same number of filters as the dimensionality of the Transformer layers. This allows more flexibility when selecting the number of attention heads for the Transformer layers since `d_model` must be evenly divisible by `n_heads`.

Interesting! Despite the wide array of different model configurations I used, the models that used the embedding layer always performed better (see the two clusters of runs on the upper right of Figure A.5). The aggregated chart in the upper left of Figure A.5 (in blue and orange) makes this pretty clear.

You can also see in Figure A.5, that according to the "Parameter Importance" measurement, the number of trainable parameters seems to be important. Maybe performance isn't dependent on whether or not an embedding is used, but rather on how many parameters the model has! I'll inspect this in the next section.

A.6 Embedding Layers, Not Just Larger Models, Improve Performance

A.6.1 Motivation

As I mentioned in the previous section, it seemed like the number of parameters in each model was more important than whether or not the model had an embedding layer. Let's try another experiment to clarify whether or not this is true.

A.6.2 Experiment

To test this hypothesis, I ran 3 models (architectures summarized in Table A.1.

- The first, `embedding-control`, has convolution and embedding layers with a Transformer layer size of 256.
- The second, `no-emb-less-params`, forgoes the embedding layer, but keeps the same Transformer layer size. As a result, there is an overall decrease in the number of parameters (from 13 million to 4 million).
- The third, `no-emb-same-params`, also forgoes the embedding layer but increases the size of the Transformer layer to 658 in an attempt to approximately match the same number of parameters as the control (about 13 million).

Again, we see in Figure A.6 that models with embedding layers (purple) perform better than other comparable models, even when we control for the number model parameters. The Parameter Importance chart in Figure A.6 supports this conclusion as well. This is interesting because the amino acid "vocabulary" is so much smaller than that of natural languages, so perhaps the embedding layer is doing something else to incorporate protein

Model Dim	Parameters	Trainable Params	Name	Embedding
648	13,275,460	13,275,460	ne2-convcomb21-11-3	No
256	13,264,920	13,264,920	m3-convcomb21-11-3	Yes
256	4,082,276	4,082,276	ne2-convcomb21-11-3	No

Table A.1: Summary of models tested in Section A.6.2.

structure information into its amino acid representations! I'll perform a quick follow-up experiment soon to visualize these embeddings. Maybe they learned something about amino acids such as their hydrophobicity or chemical properties. Who knows? Either way, I'll be sure to continue using models with embedding layers.

A.7 Conclusion

Thanks for following along with me on a typical set of experiments that I do for my research project! I hope I've been able to teach you a thing or two about protein structure as well as some of the deep learning methods researchers are using to make predictions.

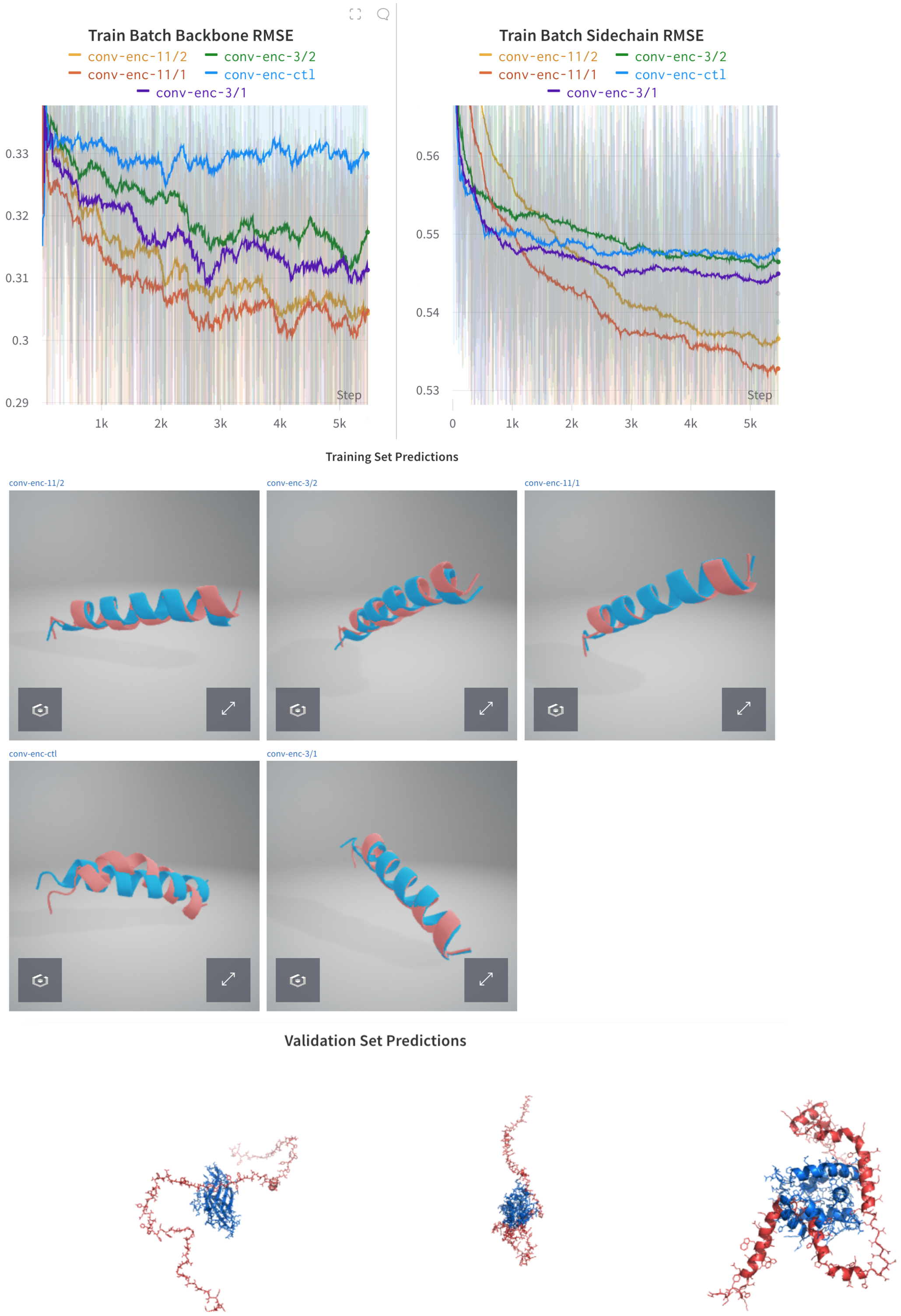


Figure A.3: Results of adding convolution layers to ProteinTransformer trained with RMSE loss. For all structure visualizations, red is the predicted structure and blue is the true structure. Both the backbone and sidechain structure elements are visible in the images labeled "Validation Set Predictions".

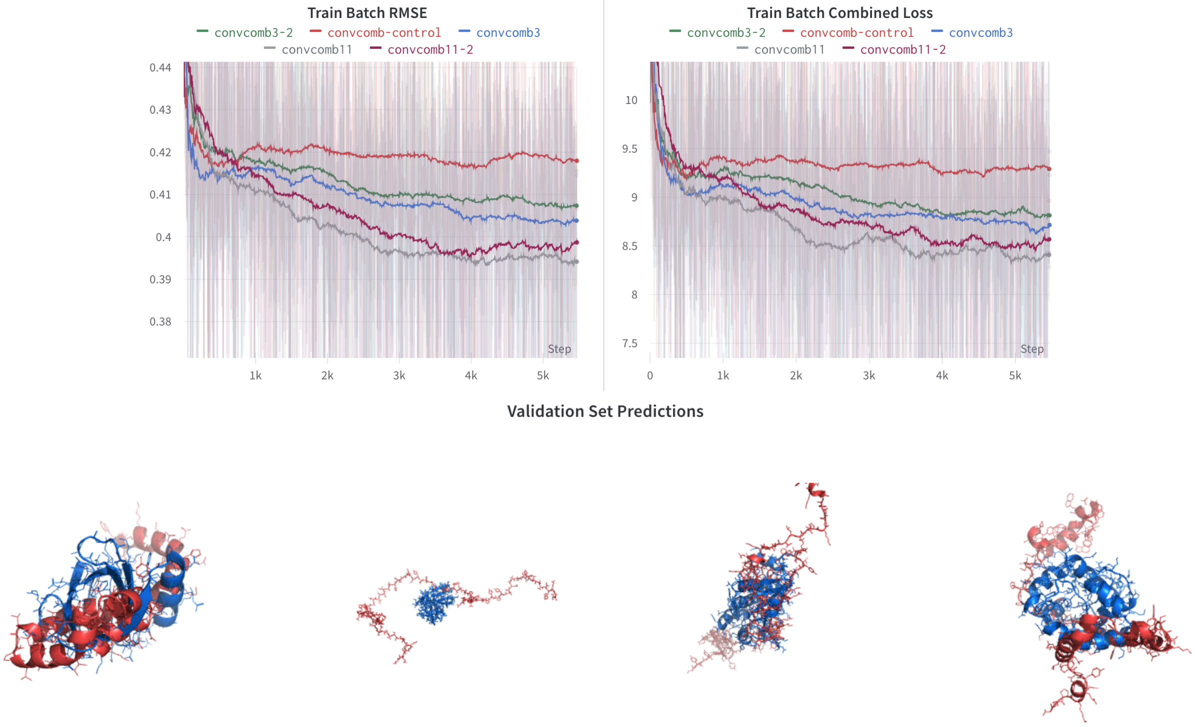
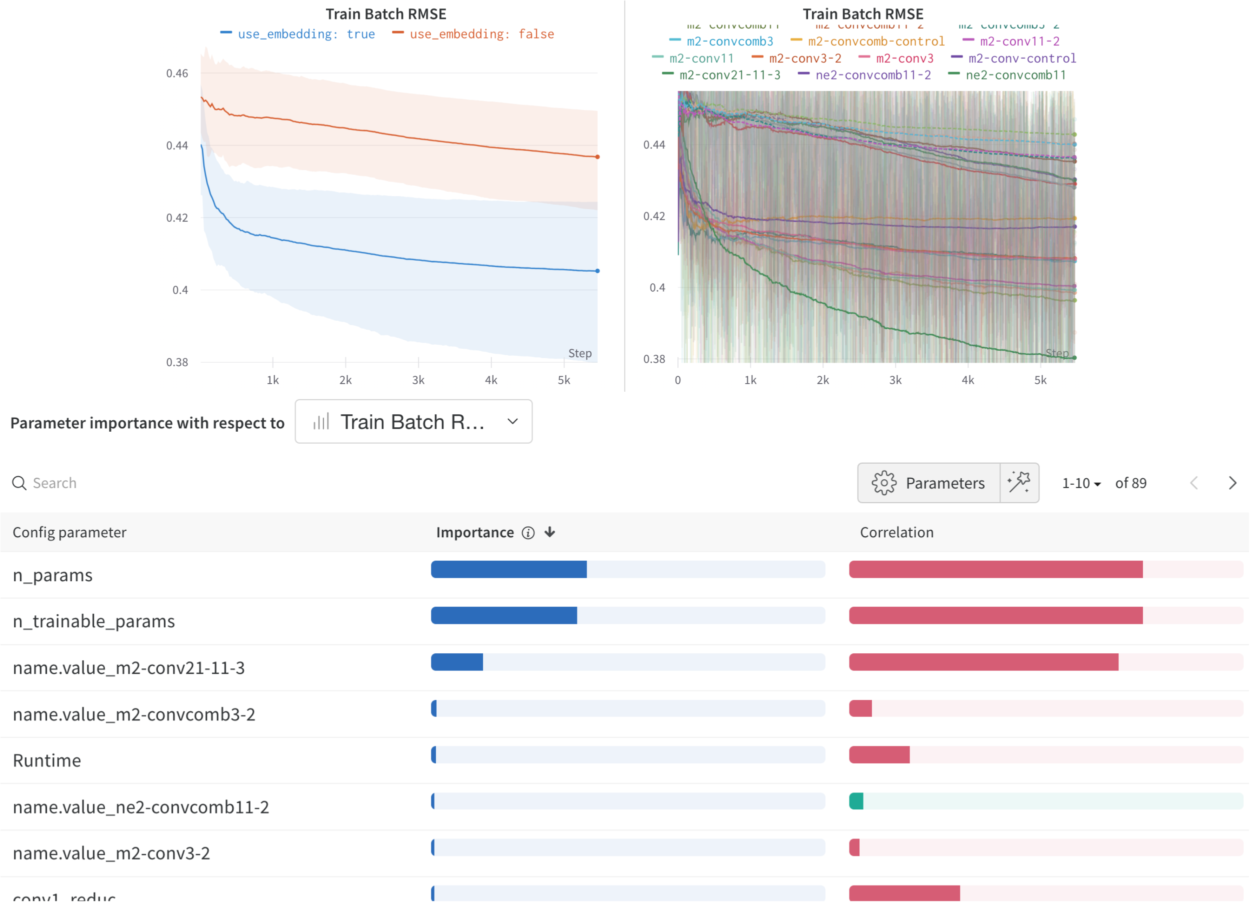


Figure A.4: Results of adding convolution layers to ProteinTransformer models trained with DRMSD loss. For all structure visualizations, red is the predicted structure and blue is the true structure. Both the backbone and sidechain structure elements are visible in the images labeled "Validation Set Predictions".



Parameter importance with respect to Train Batch R...

Search

Parameters

1-10 ▾ of 89

Config parameter	Importance ⓘ ↓	Correlation
n_params		
n_trainable_params		
name.value_m2-conv21-11-3		
name.value_m2-convcomb3-2		
Runtime		
name.value_ne2-convcomb11-2		
name.value_m2-conv3-2		
conv1_reduc		

Figure A.5: Testing the impact of embedding layers on ProteinTransformer performance.

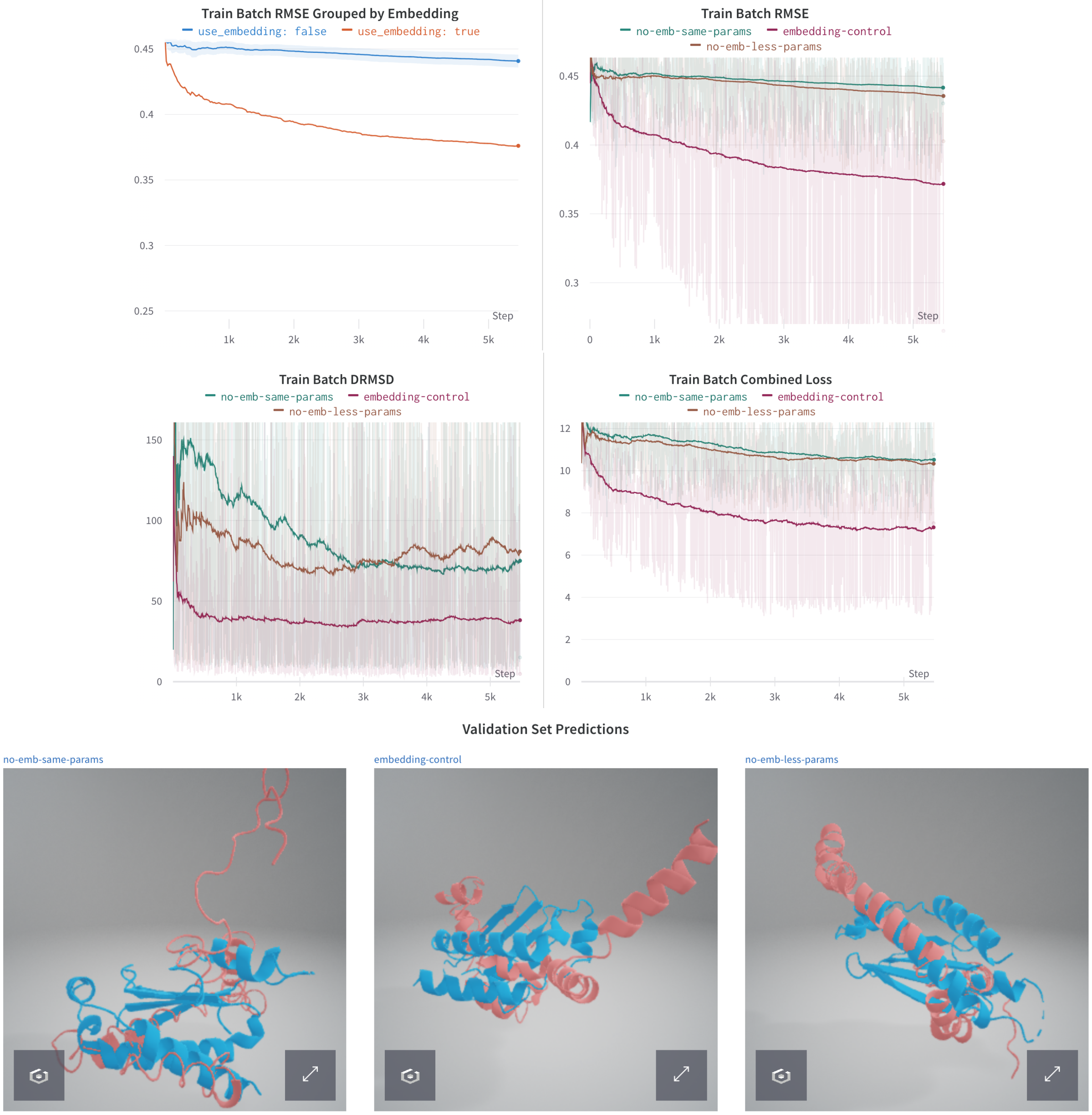
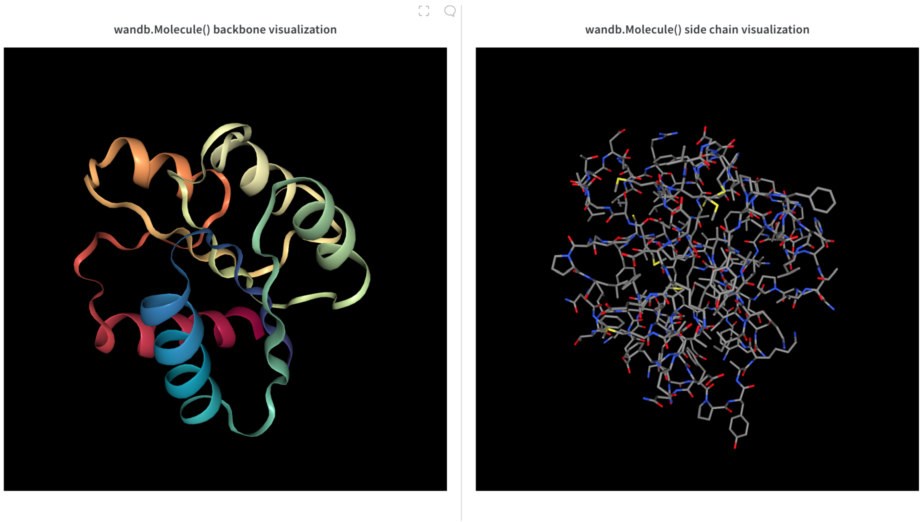


Figure A.6: Comparing the impact of embeddings on ProteinTransformers, controlling for model parameter size. For all structure visualizations, red is the predicted structure and blue is the true structure.



wandb.Molecule() backbone visualization

wandb.Molecule() side chain visualization

Figure A.7: Examples of molecular visualization tools available on Weights and Biases, thanks to Nicholas Bardy.

Bibliography

[1] Kendrew, J. C.; Bodo, G.; Dintzis, H. M.; Parrish, R. G.; Wyckoff, H.; Phillips, D. C. A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature* **1958**, *181*, 662–666, DOI: 10.1038/181662a0.

[2] Stevens, R. C. The cost and value of three-dimensional protein structure. *Drug Discovery World* **2003**, *4*, 35–48.

[3] The UniProt Consortium, UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research* **2018**, *47*, D506–D515, DOI: 10.1093/nar/gky1049.

[4] Berman, H. M. The Protein Data Bank. *Nucleic Acids Research* **2000**, *28*, 235–242, DOI: 10.1093/nar/28.1.235.

[5] HART, W. E.; ISTRAIL, S. Robust Proofs of NP-Hardness for Protein Folding: General Lattices and Energy Potentials. `https://www.liebertpub.com/doi/10.1089/cmb.1997.4.1`, Archive Location: world.

[6] Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589, DOI: 10.1038/s41586-021-03819-2, Number: 7873 Publisher: Nature Publishing Group.

[7] Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W. R.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Crossan, S.; Kohli, P.; Jones, D. T.; Silver, D.; Kavukcuoglu, K.; Hassabis, D. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706–710.

[8] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* **2015**, *115*, 211–252, DOI: 10.1007/s11263-015-0816-y.

[9] Bimbraw, K. Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO). 2015; pp 191–198.

[10] Vaswani, A.; Bengio, S.; Brevdo, E.; Chollet, F.; Gomez, A. N.; Gouws, S.; Jones, L.; Kaiser, L.; Kalchbrenner, N.; Parmar, N.; Sepassi, R.; Shazeer, N.; Uszkoreit, J. Tensor2Tensor for Neural Machine Translation. 2018.

[11] Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* **2014**, DOI: 10.3115/v1/w14-4012.

[12] Schuster, M.; Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* **1997**, *45*, 2673–2681.

[13] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.

[14] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. 2017.

[15] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**,

[16] Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Brew, J. HuggingFace's Transformers: State-of-the-art Natural Language Processing. 2019.

[17] Alammar, J. The Illustrated Transformer. `http://jalammar.github.io/illustrated-transformer/`.

[18] AlQuraishi, M. End-to-End Differentiable Learning of Protein Structure. *Cell Systems* **2019**, *8*, 292–301.e3, DOI: 10.1016/j.cels.2019.03.006.

[19] AlQuraishi, M. Some Thoughts on a Mysterious Universe. 2018; `https://moalquraishi.wordpress.com/2018/12/09/alphafold-casp13-what-just-happened/`.

[20] Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y. S. Evaluating Protein Transfer Learning with TAPE. **2019**,

[21] Hutson, M. AI protein-folding algorithms solve structures faster than ever. *Nature* **2019**, DOI: 10.1038/d41586-019-01357-6.

[22] Kryshtafovych, A.; Schwede, T.; Topf, M.; Fidelis, K.; Moult, J. Critical assessment of methods of protein structure prediction (CASP)—Round XIII. *Proteins: Structure, Function, and Bioinformatics* **2019**, *87*, 1011–1020, DOI: 10.1002/prot.25823.

[23] Pereira, J.; Simpkin, A. J.; Hartmann, M. D.; Rigden, D. J.; Keegan, R. M.; Lupas, A. N. High-accuracy protein structure prediction in CASP14. *89*, 1687–1699, DOI: https://doi.org/10.1002/prot.26171, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26171.

[24] Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods* **2019**, *16*, 1315–1322, DOI: 10.1038/s41592-019-0598-1.

[25] Ingraham, J.; Garg, V. K.; Barzilay, R.; Jaakkola, T. Generative Models for Graph-Based Protein Design. Advances in Neural Information Processing Systems. 2019.

[26] Rifaioglu, A. S.; Doğan, T.; Martin, M. J.; Cetin-Atalay, R.; Atalay, V. DEEPred: Automated Protein Function Prediction with Multi-task Feed-forward Deep Neural Networks. *Scientific Reports* **2019**, *9*, DOI: 10.1038/s41598-019-43708-3.

[27] Seo, S.; Oh, M.; Park, Y.; Kim, S. DeepFam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics* **2018**, *34*, i254–i262, DOI: 10.1093/bioinformatics/bty275.

[28] Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *118*, e2016239118, DOI: 10.1073/pnas.2016239118, Publisher: Proceedings of the National Academy of Sciences.

[29] Xu, J. Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences* **2019**, *116*, 16856–16865, DOI: 10.1073/pnas.1821309116.

[30] Yang, J.; Anishchenko, I.; Park, H.; Peng, Z.; Ovchinnikov, S.; Baker, D. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences* **2020**, *117*, 1496–1503, DOI: 10.1073/pnas.1914677117.

[31] Ingraham, J.; Riesselman, A.; Sander, C.; Marks, D. Learning Protein Structure with a Differentiable Simulator. International Conference on Learning Representations. 2019.

[32] Wu, R.; Ding, F.; Wang, R.; Shen, R.; Zhang, X.; Luo, S.; Su, C.; Wu, Z.; Xie, Q.; Berger, B.; Ma, J.; Peng, J. High-resolution de novo structure prediction from primary sequence. 2022; `https://www.biorxiv.org/content/10.1101/2022.07.21.500999v1`, Pages: 2022.07.21.500999 Section: New Results.

[33] Ratul Chowdhury,; Nazim Bouatta,; Surojit Biswas,; Charlotte Rochereau,; George M. Church,; Peter K. Sorger,; Mohammed AlQuraishi, Single-sequence protein structure prediction using language models from deep learning. 2021.08.02.454840, DOI: 10.1101/2021.08.02.454840.

[34] Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *373*, 871–876, DOI: 10.1126/science.abj8754, _eprint: https://www.science.org/doi/pdf/10.1126/science.abj8754.

[35] Jones, D. T. Setting the standards for machine learning in biology. *Nature Reviews Molecular Cell Biology* **2019**, *20*, 659–660, DOI: 10.1038/s41580-019-0176-5.

[36] Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry* **2012**, *55*, 6582–6594, DOI: 10.1021/jm300687e.

[37] Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLOS ONE* **2019**, *14*, e0220113, DOI: 10.1371/journal.pone.0220113.

[38] Sieg, J.; Flachsenberg, F.; Rarey, M. In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening. *Journal of chemical information and modeling* **2019**, *59*, 947–961.

[39] Bhuyan, M. S. I.; Gao, X. A protein-dependent side-chain rotamer library. *BMC Bioinformatics* **2011**, *12*, DOI: 10.1186/1471-2105-12-s14-s10.

[40] Shapovalov, M. V.; Dunbrack, R. L. A Smoothed Backbone-Dependent Rotamer Library for Proteins Derived from Adaptive Kernel Density Estimates and Regressions. *Structure* **2011**, *19*, 844–858, DOI: 10.1016/j.str.2011.03.019.

[41] Raman, S.; Vernon, R.; Thompson, J.; Tyka, M.; Sadreyev, R.; Pei, J.; Kim, D.; Kellogg, E.; DiMaio, F.; Lange, O.; Kinch, L.; Sheffler, W.; Kim, B.-H.; Das, R.; Grishin, N. V.; Baker, D. Structure prediction for CASP8 with all-atom refinement using Rosetta. *Proteins: Structure, Function, and Bioinformatics* **2009**, *77*, 89–99, DOI: 10.1002/prot.22540.

[42] Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Tunyasuvunakool, K.; Ronneberger, O.; Bates, R.; Žídek, A.; Bridgland, A.; Meyer, C.; A A Kohl, S.; Potapenko, A.; J Ballard, A.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Steinegger, M.; Pacholska, M.; Silver, D.; Vinyals, O.; W Senior, A.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. High Accuracy Protein Structure Prediction Using Deep Learning. 2020; `https://predictioncenter.org/casp14/doc/CASP14_Abstracts.pdf`.

[43] AlQuraishi, M. ProteinNet: a standardized data set for machine learning of protein structure. *BMC Bioinformatics* **2019**, *20*, DOI: 10.1186/s12859-019-2932-0.

[44] Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber biomolecular simulation programs. *Journal of Computational Chemistry* **2005**, *26*, 1668–1688, DOI: 10.1002/jcc.20290.

[45] Tian, C.; Kasavajhala, K.; Belfon, K. A. A.; Raguette, L.; Huang, H.; Migues, A. N.; Bickel, J.; Wang, Y.; Pincay, J.; Wu, Q.; Simmerling, C. ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *Journal of Chemical Theory and Computation* **2019**, *16*, 528–552, DOI: 10.1021/acs.jctc.9b00591.

[46] Bakan, A.; Meireles, L. M.; Bahar, I. ProDy: Protein Dynamics Inferred from Theory and Experiments. *Bioinformatics* **2011**, *27*, 1575–1577, DOI: 10.1093/bioinformatics/btr168.

[47] Schrödinger, LLC,

[48] Rego, N.; Koes, D. 3Dmol.js: molecular visualization with WebGL. *Bioinformatics* **2014**, *31*, 1322–1324, DOI: 10.1093/bioinformatics/btu829.

[49] Sillitoe, I.; Dawson, N.; Lewis, T. E.; Das, S.; Lees, J. G.; Ashford, P.; Tolulope, A.; Scholes, H. M.; Senatorov, I.; Bujan, A.; Rodriguez-Conde, F. C.; Dowling, B.; Thornton, J.; Orengo, C. A. CATH: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic Acids Research* **2018**, *47*, D280–D284, DOI: 10.1093/nar/gky1097.

[50] Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. `http://arxiv.org/abs/2006.11239`.

[51] Wang, W.; Xu, M.; Cai, C.; Miller, B. K.; Smidt, T.; Wang, Y.; Tang, J.; Gómez-Bombarelli, R. Generative Coarse-Graining of Molecular Conformations. `http://arxiv.org/abs/2201.12176`.

[52] Alcaide, E.; Biderman, S.; Telenti, A.; Maher, M. C. MP-NeRF: A Massively Parallel Method for Accelerating Protein Structure Reconstruction from Internal Coordinates. `https://www.biorxiv.org/content/10.1101/2021.06.08.446214v1`, Pages: 2021.06.08.446214 Section: New Results.

[53] Sai Pooja Mahajan,; Jeffrey A. Ruffolo,; Jeffrey J. Gray, Contextual protein encodings from equivariant graph transformers. 2023.07.15.549154, DOI: 10.1101/2023.07.15.549154.

[54] Weissenow, K.; Heinzinger, M.; Steinegger, M.; Rost, B. Ultra-fast protein structure prediction to capture effects of sequence variation in mutation movies. `https://www.biorxiv.org/content/10.1101/2022.11.14.516473v2`, Pages: 2022.11.14.516473 Section: New Results.

[55] Avery, C.; Patterson, J.; Grear, T.; Frater, T.; Jacobs, D. J. Protein Function Analysis through Machine Learning. *12*, 1246, DOI: 10.3390/biom12091246, Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[56] Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, *379*, 1123–1130, DOI: 10.1126/science.ade2574, Publisher: American Association for the Advancement of Science.

[57] Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research* **2000**, *28*, 235–242, DOI: 10.1093/nar/28.1.235.

[58] Masha, K.; J, N. J.; O, D. R. How accurately can one predict drug binding modes using AlphaFold models? *eLife* **2023**, *12*, DOI: 10.7554/eLife.89386, Publisher: eLife Sciences Publications Limited.

[59] Scardino, V.; Di Filippo, J. I.; Cavasotto, C. N. How good are AlphaFold models for docking-based virtual screening? *iScience* **2022**, *26*, 105920, DOI: 10.1016/j.isci.2022.105920.

[60] He, X.-h.; You, C.-z.; Jiang, H.-l.; Jiang, Y.; Xu, H. E.; Cheng, X. AlphaFold2 versus experimental structures: evaluation on G protein-coupled receptors. *Acta Pharmacologica Sinica* **2023**, *44*, 1–7, DOI: 10.1038/s41401-022-00938-y, Number: 1 Publisher: Nature Publishing Group.

[61] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36, DOI: 10.1021/ci00057a005, Publisher: American Chemical Society.

[62] Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design* **2016**, *30*, 595–608, DOI: 10.1007/s10822-016-9938-8.

[63] Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. 2015; `http://arxiv.org/abs/1509.09292`, arXiv:1509.09292 [cs, stat].

[64] Wang, Y.; Wu, S.; Duan, Y.; Huang, Y. A Point Cloud-Based Deep Learning Strategy for Protein-Ligand Binding Affinity Prediction. 2021; `http://arxiv.org/abs/2107.04340`, arXiv:2107.04340 [q-bio].

[65] McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: molecular docking with deep learning. *Journal of Cheminformatics* **2021**, *13*, 43, DOI: 10.1186/s13321-021-00522-2.

[66] McNutt, A. T.; Koes, D. R. Improving G Predictions with a Multitask Convolutional Siamese Network. *Journal of Chemical Information and Modeling* **2022**, *62*, 1819–1829, DOI: 10.1021/acs.jcim.1c01497, Publisher: American Chemical Society.

[67] Lookman, T.; Balachandran, P. V.; Xue, D.; Yuan, R. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials* **2019**, *5*, 1–17, DOI: 10.1038/s41524-019-0153-8, Number: 1 Publisher: Nature Publishing Group.

[68] Khalak, Y.; Tresadern, G.; Hahn, D. F.; de Groot, B. L.; Gapsys, V. Chemical Space Exploration with Active Learning and Alchemical Free Energies. *Journal of Chemical Theory and Computation* **2022**, *18*, 6259–6270, DOI: 10.1021/acs.jctc.2c00752, Publisher: American Chemical Society.

[69] Francoeur, P.; Penaherrera, D.; Koes, D. Active Learning for Small Molecule pKa Regression; a Long Way To Go. 2022; `https://chemrxiv.org/engage/chemrxiv/article-details/6286a54759f0d637bb968d0d`.

[70] Ahdritz, G.; Bouatta, N.; Kadyan, S.; Xia, Q.; Gerecke, W.; O'Donnell, T. J.; Berenberg, D.; Fisk, I.; Zanichelli, N.; Zhang, B.; Nowaczynski, A.; Wang, B.; Stepniewska-Dziubinska, M. M.; Zhang, S.; Ojewole, A.; Guney, M. E.; Biderman, S.; Watkins, A. M.; Ra, S.; Lorenzo, P. R.; Nivon, L.; Weitzner, B.; Ban, Y.-E. A.; Sorger, P. K.; Mostaque, E.; Zhang, Z.; Bonneau, R.; AlQuraishi, M. OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. 2022; `https://www.biorxiv.org/content/10.1101/2022.11.20.517210v2`, Pages: 2022.11.20.517210 Section: New Results.

[71] King, J. E.; Koes, D. R. SidechainNet: An all-atom protein structure dataset for machine learning. *Proteins: Structure, Function, and Bioinformatics* **2021**, *89*, 1489–1496, DOI: 10.1002/prot.26169, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26169.

[72] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 2019; `http://arxiv.org/abs/1912.01703`, arXiv:1912.01703 [cs, stat].

[73] Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Computational Biology* **2017**, *13*, e1005659, DOI: 10.1371/journal.pcbi.1005659.

[74] Debiec, K. T.; Cerutti, D. S.; Baker, L. R.; Gronenborn, A. M.; Case, D. A.; Chong, L. T. Further along the Road Less Traveled: AMBER ff15ipq, an Original Protein Force Field Built on a Self-Consistent Physical Model. *Journal of Chemical Theory and Computation* **2016**, *12*, 3926–3947, DOI: 10.1021/acs.jctc.6b00567, Publisher: American Chemical Society.

[75] Parsons, J.; Holmes, J. B.; Rojas, J. M.; Tsai, J.; Strauss, C. E. M. Practical conversion from torsion space to Cartesian space for in silico protein synthesis. *Journal of Computational Chemistry* **2005**, *26*, 1063–1068, DOI: 10.1002/jcc.20237, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.20237.

[76] Milanesi, L.; Waltho, J. P.; Hunter, C. A.; Shaw, D. J.; Beddard, G. S.; Reid, G. D.; Dev, S.; Volk, M. Measurement of energy landscape roughness of folded and unfolded proteins. *Proceedings of the National Academy of Sciences* **2012**, *109*, 19563–19568, DOI: 10.1073/pnas.1211764109, Publisher: Proceedings of the National Academy of Sciences.

[77] Gruebele, M. Protein folding: the free energy surface. *Current Opinion in Structural Biology* **2002**, *12*, 161–168, DOI: 10.1016/S0959-440X(02)00304-4.

[78] Schaarschmidt, J.; Monastyrskyy, B.; Kryshtafovych, A.; Bonvin, A. M. Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age. *Proteins:*

Structure, Function, and Bioinformatics **2018**, *86*, 51–66, DOI: 10.1002/prot.25407, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25407.

[79] Steinegger, M.; Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology* **2017**, *35*, 1026–1028, DOI: 10.1038/nbt.3988, Number: 11 Publisher: Nature Publishing Group.

[80] Robin, X.; Haas, J.; Gumienny, R.; Smolinski, A.; Tauriello, G.; Schwede, T. Continuous Automated Model EvaluatiOn (CAMEO)—Perspectives on the future of fully automated evaluation of structure prediction methods. *Proteins: Structure, Function, and Bioinformatics* **2021**, *89*, 1977–1986, DOI: 10.1002/prot.26213, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26213.

[81] Mariani, V.; Biasini, M.; Barbato, A.; Schwede, T. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* **2013**, *29*, 2722–2728, DOI: 10.1093/bioinformatics/btt473.

[82] Williams, C. J.; Headd, J. J.; Moriarty, N. W.; Prisant, M. G.; Videau, L. L.; Deis, L. N.; Verma, V.; Keedy, D. A.; Hintze, B. J.; Chen, V. B.; Jain, S.; Lewis, S. M.; Arendall III, W. B.; Snoeyink, J.; Adams, P. D.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C. MolProbity: More and better reference data for improved all-atom structure validation. *Protein Science* **2018**, *27*, 293–315, DOI: 10.1002/pro.3330, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.3330.

[83] Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **2020**, *17*, 261–272, DOI: 10.1038/s41592-019-0686-2, Number: 3 Publisher: Nature Publishing Group.

[84] Clark, J. J.; Benson, M. L.; Smith, R. D.; Carlson, H. A. Inherent versus induced protein flexibility: Comparisons within and between apo and holo structures. *PLoS Computational Biology* **2019**, *15*, e1006705, DOI: 10.1371/journal.pcbi.1006705.

[85] Gaudreault, F.; Chartier, M.; Najmanovich, R. Side-chain rotamer changes upon ligand binding: common, crucial, correlate with entropy and rearrange hydrogen bonding. *28*, i423–i430, DOI: 10.1093/bioinformatics/bts395.

[86] Krivov, G. G.; Shapovalov, M. V.; Dunbrack, R. L. Improved prediction of protein side-chain conformations with SCWRL4. *77*, 778–795, DOI: 10.1002/prot.22488.

[87] Huang, X.; Pearce, R.; Zhang, Y. FASPR: an open-source tool for fast and accurate protein side-chain packing. *36*, 3758–3765, DOI: 10.1093/bioinformatics/btaa234.

[88] McPartlon, M.; Xu, J. An end-to-end deep learning method for rotamer-free protein side-chain packing. `https://www.biorxiv.org/content/10.1101/2022.03.11.483812v1`, Pages: 2022.03.11.483812 Section: New Results.

[89] Xu, G.; Wang, Q.; Ma, J. OPUS-Rota4: a gradient-based protein side-chain modeling framework assisted by deep learning-based predictors. *23*, bbab529, DOI: 10.1093/bib/bbab529.

[90] Misiura, M.; Shroff, R.; Thyer, R.; Kolomeisky, A. B. DLPacker: Deep learning for prediction of amino acid side chain conformations in proteins. *90*, 1278–1290, DOI: 10.1002/prot.26311, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26311.

[91] Chu, A. E.; Cheng, L.; Nesr, G. E.; Xu, M.; Huang, P.-S. An all-atom protein generative model. https://www.biorxiv.org/content/10.1101/2023.05.24.542194v1, Pages: 2023.05.24.542194 Section: New Results.

[92] Heo, L.; Janson, G.; Feig, M. Physics-based protein structure refinement in the era of artificial intelligence. *89*, 1870–1887, DOI: 10.1002/prot.26161.

[93] Zhang, Y.; Vass, M.; Shi, D.; Abualrous, E.; Chambers, J.; Chopra, N.; Higgs, C.; Kasavajhala, K.; Li, H.; Nandekar, P.; Sato, H.; Miller, E.; Repasky, M.; Jerome, S. Benchmarking Refined and Unrefined AlphaFold2 Structures for Hit Discovery. https://chemrxiv.org/engage/chemrxiv/article-details/62b41f0c0bbbc117477285a4.

[94] Wu, T.; Guo, Z.; Cheng, J. Atomic protein structure refinement using all-atom graph representations and SE(3)-equivariant graph transformer. *39*, btad298, DOI: 10.1093/bioinformatics/btad298.

[95] Adiyaman, R.; Edmunds, N. S.; Genc, A. G.; Alharbi, S. M. A.; McGuffin, L. J. Improvement of protein tertiary and quaternary structure predictions using the ReFOLD refinement method and the AlphaFold2 recycling process. *3*, vbad078, DOI: 10.1093/bioadv/vbad078.

[96] Liu, J.; Guo, Z.; Wu, T.; Roy, R. S.; Chen, C.; Cheng, J. Improving AlphaFold2-based protein tertiary structure prediction with MULTICOM in CASP15. *6*, 188, DOI: 10.1038/s42004-023-00991-6.

[97] Wu, F.; Jing, X.; Luo, X.; Xu, J. Improving protein structure prediction using templates and sequence embedding. *39*, btac723, DOI: 10.1093/bioinformatics/btac723.

[98] Oda, T. Refinement of AlphaFold-Multimer structures with single sequence input. https://www.biorxiv.org/content/10.1101/2022.12.27.521991v2, Pages: 2022.12.27.521991 Section: New Results.

[99] Yang, K. K.; Fusi, N.; Lu, A. X. Convolutions are competitive with transformers for protein sequence pretraining. https://www.biorxiv.org/content/10.1101/2022.05.19.492714v4, Pages: 2022.05.19.492714 Section: New Results.

[100] King, J. Evaluating the Impact of Sequence Convolutions & Embeddings on Protein Structure Prediction. https://wandb.ai/koes-group/protein-transformer/reports/Evaluating-the-Impact-of-Sequence-Convolutions-Embeddings-on-Protein-Structure-Prediction--Vmlldzo2OTg4Nw.

[101] GORBI: Gene Ontology at Rudjer Boskovic Institute :: Growth of sequence databases. http://gorbi.irb.hr/en/method/growth-of-sequence-databases/.