Evidence for and Applications of Physics-Based Reasoning in AlphaFold

Citation

Permanent link
https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37371758

Terms of Use
This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

Share Your Story
The Harvard community has made this article openly available. Please share how this access benefits you. Submit a story.

Accessibility
Evidence for and Applications of Physics-Based Reasoning in AlphaFold

A THESIS PRESENTED
BY
JAMES RONEY
TO
THE DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ARTS
IN THE SUBJECTS OF
COMPUTER SCIENCE

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
MARCH 2022
Evidence for and Applications of Physics-Based Reasoning in AlphaFold

Abstract

The problem of predicting a protein’s 3D structure from its primary amino acid sequence is a longstanding challenge in structural biology. Recently, approaches like AlphaFold have achieved remarkable performance on this task by using deep learning techniques to analyze patterns of variation across evolutionarily related protein sequences. The use of such co-evolutionary information is critical to these models’ accuracy, and without it their predictive performance drops considerably. In living cells, however, the 3D structure of a protein is fully determined by its primary sequence and the biophysical laws that cause it to fold into a low-energy configuration. Thus, it should be possible to predict a protein’s structure from only its primary sequence by learning a highly-accurate biophysical energy function. We provide evidence that AlphaFold has learned such an energy function, and uses coevolution data to solve the global search problem of finding a low-energy conformation. We demonstrate that AlphaFold’s learned potential function can be used to rank the quality of candidate protein structures with state-of-the-art accuracy, without using any coevolution data. Finally, we explore practical applications of this learned potential function, including predicting protein structures without coevolution data and predicting the effects of mutations on proteins. By iteratively optimizing protein structures using AlphaFold’s learned potential function, we are able to create significantly improved protein structure predictions without the use of coevolution information, which represents an important step toward the goal of predicting protein structures from single sequences using physical principles.
## Contents

1 **INTRODUCTION**  
1.1 Background and Motivations .............................................. 1  
1.2 Our Contributions .......................................................... 4  
1.3 Roadmap ................................................................. 5  

2 **PROTEINS: EVOLUTION, STRUCTURE, AND FUNCTION**  
2.1 Fundamentals of Protein Biology .................................. 6  
2.2 Protein Structure and Function .................................... 13  
2.3 Experimental Determination of Protein Structures ............ 22  
2.4 Evolutionary Relationships Between Proteins .................. 25  

3 **PROTEIN STRUCTURE PREDICTION: A GRAND CHALLENGE**  
3.1 Preliminaries and Metrics ......................................... 29  
3.2 Physics-Based Methods ............................................... 32  
3.3 Coevolution: Biodiversity Reveals Structure .................... 35  
3.4 Contributions from Deep Learning ................................. 39  
3.5 AlphaFold: A Breakthrough ........................................ 45  
3.6 Remaining Challenges .............................................. 48  

4 **A LEARNED BIOPHYSICAL POTENTIAL IN ALPHAFOLD**  
4.1 Interpretability in AlphaFold: Questions and Hypotheses .... 50  
4.2 Ranking Decoy Structures with AlphaFold ....................... 51  
4.3 Discussion ................................................................. 67  

5 **APPLICATIONS OF PHYSICS-BASED REASONING IN ALPHAFOLD**  
5.1 MSA-Free Structure Prediction ..................................... 69  
5.2 Mutation Effect Prediction ......................................... 73
5.3 Discussion ................................................................. 76

6 Conclusions ......................................................... 78

References ............................................................. 87
To Mom, Dad, and Anna
Acknowledgments

First and foremost, thank you to Dr. Sergey Ovchinnikov for being a great advisor on this project. Your mentorship on this project has helped convince me that I want to study the world of proteins in graduate school and beyond. Thank you to Henri Palacci, Paul Maragakis, and Peter Skopp at D.E. Shaw Research for first sparking my interest in computational protein science – I look forward to working together in the coming years. I’m also thankful to Dr. John Jumper of Deepmind, both for his inspiring work on AlphaFold and his helpful comments on a recent preprint containing some of this work. Finally, I’d like to give special thanks to the numerous fantastic teachers and mentors who have fostered my curiosity over the years – especially Jim Pearson, Paul Campbell, Kim Miller, Lennie Araki, Andrew Duerner, Michael Goard, Joe Blitzstein, James Mickens, Franziska Michor, Ollie McDonald, and Justin Dean.
1

Introduction

1.1 Background and Motivations

Proteins are the molecular workhorses of life. These incredible molecules are responsible for conducting electric currents in our brains, contracting our muscles, fighting off diseases, detecting light, sounds, smells, tastes, heat, and vibrations, reading and replicating our genetic code, metabolizing our food, signalling between our cells, and much more. Essentially every biological process on earth is carried out by specialized proteins [18].

Despite the enormous diversity of functions carried out by proteins, they have a very simple, stereotyped chemical structure. Each protein is a linear chain of the same 20 chemical building blocks known as amino acids. One sequence of amino acids creates a protein that detects light in the human eye, and another ordering of the same 20 building blocks creates the spike protein used by the coronavirus to enter human cells. These protein sequences are encoded into organisms’ DNA, and vary over the course of evolution due to random mutations and the pressures of natural selection.

How is it possible for such a wide range of functions to arise from such a small number of chemical components? The answer lies in the fact that proteins are not structurally uniform. Rather, sequences of amino acids fold up to form complex and irregular 3D structures, and those structures are crucial for giving proteins their specific functions. For ex-
ample, proteins that transport materials in and out of cells are shaped like microscopic tubes, while the spike protein has an elongated shape that allows it to protrude from the virus and dock with proteins on the surfaces of cells (Figure 1.1). These examples are simplifications, and in general the link between structure and function is too complex to be apparent from visual inspection alone. Nonetheless, the essential idea that unique structures give proteins their unique functions is a foundational principle of protein biology [18].

![A.](image1.png) ![B.](image2.png)

**Figure 1.1:** Two example protein structures. A.) A porin protein, which creates a channel for materials to move in and out of the cell. PDB 2POR. B.) The SARS-CoV-2 spike protein. PDB 7FCD.

Determining the structures of proteins is of great scientific interest. Because of the intimate relationship between protein structures and functions, it is often necessary to figure out a protein’s shape in order to understand how it accomplishes its biological task. Determining protein structures is also of great practical importance for many applications. For example, in order to design drugs that interact with disease-causing proteins, it is often necessary to know the structures of those target proteins. Protein structures are typically determined through careful experiments, the details of which will be covered later in this thesis.

Often times, the experiments needed to determine protein structures are time and labor-
intensive [25, 27]. This can be an especially relevant issue in situations like the COVID-19 pandemic: A novel virus containing dozens of unknown protein structures was discovered, all of which were relevant to understanding the disease biology of COVID-19 and developing vaccines and therapeutics. Structural biologists worked quickly to determine these protein structures, which were instrumental in the rapid development of effective vaccines and drugs [22]. Still, this situation highlights the utility of computational methods to predict the 3D structures of proteins from their amino acid sequences, thus reducing the need for difficult experiments and saving precious time in high-stakes situations. The computational prediction of protein structures should be possible in theory, since the folded structure of a protein is fully determined by its amino acid sequence.

Recently, the machine learning model AlphaFold achieved breakthrough performance on the task of protein structure prediction, with predictions often nearing experimental accuracy [16]. To predict the structure of a target amino acid sequence, AlphaFold first searches a large database of protein sequences to compile a Multiple Sequence Alignment (MSA), which is essentially a collection of sequences that are evolutionarily related to the target sequence. MSAs are known to provide extremely useful information for predicting protein structures [3, 13, 29, 43]. Intuitively, if two amino acids are in contact in the folded protein structure, mutations in the first position may induce a selective pressure for the second position to mutate. Such mutational covariance can be detected in MSAs, and this rich signal has been critical to the success of many recent protein structure prediction models, including AlphaFold. While AlphaFold is technically capable of predicting structures without MSAs, its predictive accuracy drops dramatically without coevolution information.

The requirement of MSAs for protein structure prediction is sometimes problematic. For proteins with no known related sequences, the lack of coevolutionary information makes structure prediction difficult, potentially limiting our ability to predict the structures of very novel proteins. In theory, it should be possible to predict protein structures from amino acid sequences with no MSAs, since protein structures are fully determined by their amino acid sequences and the laws of physics that govern their folding. In particular, protein structures fold to minimize free energy, which is a function of the protein’s 3D configuration and its amino acid sequence [2, 18]. Therefore, if one could model this free energy function with sufficient accuracy, then one could predict protein structures by optimiz-
ing this function over the space of 3D configurations for a given sequence. The challenges with this approach are twofold. First, accurately characterizing the biophysical potential function that governs protein folding in a way that is computationally tractable is a very complex and difficult challenge. Second, even if one had perfect knowledge of the potential function, there are an astronomically large number of possible protein geometries, so searching for the optimum is a very difficult optimization task. The apparent impossibility of searching this vast configuration space is known as Levinthal’s Paradox [46].

Given the theoretical possibility of predicting protein structures without MSAs, it is interesting to speculate why AlphaFold remains dependent on MSAs for its accuracy. One intriguing possibility is that AlphaFold has learned an accurate potential function for scoring the accuracy of candidate protein structures, but the coevolutionary information in the MSA is necessary to locate an approximate global minimum in this potential function and circumvent the challenge of Levinthal’s Paradox. After finding the neighborhood of the global minimum using the MSA, the later stages of the AlphaFold model may act as an “unrolled optimizer” and locally descend the learned potential to produce a refined structure prediction. AlphaFold also outputs a variety of confidence scores related to the predicted accuracy of its structures, and these confidence scores may be determined by the value of its internal potential function.

This hypothesis is theoretically appealing, and its veracity has important implications for outstanding problems in computational protein science. If AlphaFold has learned an accurate potential function that does not depend on MSAs, this opens new opportunities for accurately predicting protein structures without using coevolution data. We have hypothesized that AlphaFold uses MSAs to intelligently sample a starting point for optimizing the learned potential function. However, it may be possible to replace this mechanism with a generative model that repeatedly samples starting points, thereby eliminating the need for the MSA.

1.2 Our Contributions

In this thesis, we experimentally test this hypothesis, and confirm that AlphaFold has learned accurate biophysical potential function which does not depend on MSAs. This potential function can be used to rank the accuracy of candidate protein structures without
the need for coevolution information, and it significantly outperforms previously designed physical potential functions at this task. This finding demonstrates that AlphaFold has learned important principles behind the physics of protein folding, which was not previously clear.

Subsequently, we explore applications of AlphaFold’s learned biophysical potential function. Given the accuracy of AlphaFold’s learned potential function, it should be possible to predict highly accurate structures for a given amino acid sequence by searching for optima in this potential, without the need for any coevolution information. This is important, as it could enable the prediction of protein structures for sequences with little evolutionary data. We demonstrate that this is indeed feasible, and use this approach to generate structure predictions without the need for MSAs; these predictions are significantly more accurate than the MSA-free structures produced by AlphaFold’s default prediction workflow. This exciting result is an important step towards the goal of accurately predicting protein structures from single sequences based on physical principles. In addition, we explore the application of AlphaFold’s learned potential function to predicting the effects of mutations on protein structures and functions.

1.3 Roadmap

We begin by reviewing the essential chemistry, biology, and physics of proteins and protein folding in Chapter 2. Chapter 3 then discusses the problem of protein structure prediction, including the insights and approaches that are key to the success of AlphaFold. The original contributions of this thesis begin in Chapter 4, which presents evidence for a learned biophysical potential function in AlphaFold. Applications of this potential function are explored in Chapter 5, and Chapter 6 concludes by proposing future research directions.
2.1 FUNDAMENTALS OF PROTEIN BIOLOGY

This thesis explores computational methods for predicting, analysing, and representing protein structures. In order to motivate this work, it is helpful to review the fundamentals of protein biology and chemistry.

2.1.1 ATOMIC AND MOLECULAR INTERACTIONS

Proteins are a family of biological macromolecules, and like all molecules, they are composed of atoms. Molecules and atoms interact in a variety of different ways, and these different types of bonds and interactions are key to determining the structural properties of proteins. A summary of relevant types of atomic bonds and molecular interactions is presented in Figure 2.1.

In organic molecules, atoms are primarily held together by covalent bonds, in which bonded atoms share one or more pairs of electrons. Covalent bonds are sufficiently strong that, when reasoning about protein structures, we can envision covalent bonds as a set of rigid, unchanging links that hold bonded atoms close together. It is important to note that, in covalent bonds where a single pair of electrons is shared, the bonded atoms can
Figure 2.1: Important types of interactions between atoms and molecules. A.) Single covalent bonds are free to rotate. B.) Double and triple bonds are not free to rotate. C.) Charged ions can form strong ion-ion interactions (salt bridges). D.) Dipole-dipole interactions form between polar molecules. Hydrogen bonds are a type of dipole-dipole interaction. E.) Van der Waals interactions create attractions between nonpolar molecules due to transient asymmetries in charge distribution.

rotate around the bond axis. In contrast, double and triple bonds generally cannot rotate. This effect is illustrated in Figure 2.1A and 2.1B.

Ionic bonds are another important type of interaction between atoms. These interactions occur when a positively charged ion (such as Na\(^+\)) and a negatively charged ion (such as Cl\(^-\)) are electrostatically attracted to each other, and come together to form a molecule like NaCl (table salt). Such compounds are generally referred to as “salts,” and ionic bonds can be referred to as “salt-bridges.” An ionic bond is illustrated in Figure 2.1C.
Charged ions are not the only source of electrostatic attractions between atoms and molecules. In many covalent bonds, charge is not distributed evenly between the bonded atoms. For example, in the O-H bonds of the molecule H$_2$O (water), electrons are more concentrated around the oxygen atom than the hydrogen. As a result, the oxygen atom of H$_2$O has a partial negative charge, while the hydrogens have a partial positive charge. We say that H$_2$O molecule is polar, or that it is a dipole. This happens because oxygen has a higher electronegativity than hydrogen, which is essentially a propensity to attract electrons away from other atoms. For the purposes of understanding protein structure, it is sufficient to remember that nitrogen and oxygen are highly electronegative. Carbon and hydrogen are less electronegative, although carbon is more electronegative than hydrogen.

Because dipoles have partial charges, they can form electrostatic attractions with one another. These dipole-dipole interactions are like weaker forms of salt bridges. Dipoles are also attracted to charged ions, and ion-dipole interactions are of intermediate strength between salt bridges and dipole-dipole interactions. Since water is polar, water molecules are attracted to any ion placed in an aqueous solution. This reduces the strength of salt bridges in water relative to their strength in a vacuum, and allows salts to be dissolved in water.

One especially important type of dipole-dipole interaction is a hydrogen bond. As in the H$_2$O example above, strong dipoles are often formed when an electronegative atom is bonded to hydrogen. If a hydrogen-containing dipole interacts with another dipole, the partially positive hydrogen will be attracted to the electronegative atom of the other dipole. The first dipole is called a hydrogen bond donor, while the second is an acceptor. Hydrogen bonds are ubiquitous in biological molecules, including proteins, and water molecules constantly form hydrogen bonds with one another. An example of a hydrogen bond is given in Figure 2.1D.

Finally, electrostatic attractions can occur even between non-polar, neutral molecules. To see how, note that electrons can be thought of as being dispersed in a “cloud” around a molecule. Even if a molecule is non-polar, its electron cloud may temporarily exist in an asymmetric distribution by chance, resulting in a slight polarization. This slight polarization can induce the electrons of a nearby neutral molecule to adopt a complementary polarization. These asymmetries in electron distribution mutually reinforce one another, resulting in a weak electrostatic attraction between the two. These interactions are called
Van der Waals interactions, and are illustrated in Figure 2.1E.

Note that, while Van der Waals interactions can pull atoms together, they also create repulsions between atoms that become too close. Once two atoms become close enough that the negative charges of their respective electron clouds strongly repel, they are said to have reached their Van der Walls radii. The net result of attractive and repulsive Van der Waals forces is that electrostatically interacting atoms and molecules settle at nonzero distances from one another.

As a consequence of the types of bonds described above, various compounds can be described as “hydrophillic” (water loving) and “hydrophobic” (water fearing). Charged or polar molecules are hydrophillic. Water molecules are attracted to these compounds because they can form dipole-dipole or ion-dipole interactions with them, and as a result these compounds dissolve easily in water. Hydrophobic molecules are neutral and nonpolar. When introduced to water, water molecules are more attracted to other water molecules than these compounds. As a result, these compounds will segregate to minimize mixing with the water, as happens when oil is introduced to water [18].

2.1.2 The Chemical Structure of Proteins

As hinted at in the introduction to this thesis, proteins can adopt an enormous array of shapes, appearances, and functions. However, the chemical composition of proteins is surprisingly simple to describe: Every protein in every living organism is a linear, covalently-bonded chain of 20 “building block” molecules called amino acids. The diversity of protein structure and function comes from the exponentially large number of possible sequences of amino acid building blocks – for an average-length protein of 300 amino acids, there are astronomical $20^{300}$ possible sequences.

All 20 amino acids have a common chemical form, which is depicted in Figure 2.2A. To summarize, amino acids are built around a central carbon atom, which we denote as Cα. Two charged chemical groups are attached to the Cα atom: an amino (NH$_2^+$) group, and a carboxylate group (COO$^-$) group. In addition to these two groups, the Cα atom is bonded to a hydrogen, as well as a variable side chain (denoted R in Figure 2.2A). The amino acids found in proteins differ only in the chemical composition of their side chains, and there are 20 different side chains found in proteins. The various side chains have dif-
ferent chemical properties, and it is these chemical differences that allow for such a great diversity of form and function in proteins [18].

![Chemical structures of amino acids](image)

**Figure 2.2:** The chemical structure of proteins. A.) The general structure of an amino acid B.) Various example amino acids and the properties of their side chains.

Figure 2.2B gives examples of various amino acid side chains and their properties. For the purposes of this thesis, it is sufficient to understand several broad categories of amino acid side chains. Some amino acid side chains are charged. Two amino acids (Gultamate and Aspartate) are always negatively charged, another two (Arginine and Lysine) are always positively charged, and Histidine is sometimes positively charged depending on the surrounding environment. In accordance with the previous section, these amino acids are hydrophillic. Another five amino acids are polar, making them hydrophillic as well. The remaining amino acids are nonpolar and therefore hydrophobic, which has important implications for protein structure and folding. While it is not necessary to know the exact chemical structure of the amino acid side chains for this thesis, it will be useful to note that, in 19 out of the 20 amino acids, the “first” atom in the side chain is a carbon bonded to the C\(\alpha\) atom. This carbon is denoted as C\(\beta\). Glycine is the only amino acid with no C\(\beta\) atom, as its entire side chain is simply a single hydrogen bonded to C\(\alpha\). Alanine is the next-simplest amino acid, as its side chain consists of the C\(\beta\) atom and no other heavy
Amino acids are connect to form proteins via covalent peptide bonds. As illustrated in Figure 2.3, peptide bonds are single covalent bonds between the carboxylate carbon of one amino acid and the amino nitrogen of another amino acid. At one end of the chain there is a free NH$_3^+$ group, and at the other there is a free COO$^-$ group. These ends are referred to as the N- and C-terminus, respectively. To represent a protein sequence, biologists use a set of one-letter abbreviations for each amino acid, and write the sequence from the N-terminus to the C-terminus. An example is given is Figure 4c. Note that, after peptide bonding, the polypeptide chain no longer has amino or carboxylate groups, except at the termini. For this reason, amino acids connected by peptide bonds are referred to as amino acid residues, since they no longer possess the functional groups after which amino acids are named.

![Peptide Bond](image)

**Figure 2.3:** A short polypeptide chain, written from N-terminus to C-terminus.

### 2.1.3 The Central Dogma of Molecular Biology

So far, we have reviewed the basic principles of interactions between atoms and molecules, and we have described the chemical structure of proteins. Our introduction hinted that proteins play all sorts of critical biological functions, but we have not yet described how they are made, or how living organisms “know” how to create proteins with the right sequences to perform different tasks. The profound answers to these questions are given by the central dogma of molecular biology:
The central dogma states that DNA is the information-carrying molecule of life, and the replication of DNA between generations of cells and organisms allows for the inheritance of traits. Inside living cells, portions of the information contained in DNA are transferred to RNA in a process called transcription. Finally, the information contained in RNA molecules is decoded into protein sequences in a process called translation. These proteins, whose sequences are directly encoded in DNA, then perform a myriad of functions within the cell, and form the molecular basis for an organism’s phenotypic traits.

DNA (short for deoxyribonucleic acid) is at the heart of the central dogma. Like proteins, DNA is a biological macromolecule that is built from a covalently bonded chain of molecular subunits. These subunits are called nucleotides, and there are four of them that make up DNA: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). The sequence of these four “letters” in an organism’s DNA encode all of its genetic information. The exact chemical structure of DNA is not important for this thesis, but it should be noted that a DNA sequence has two distinct ends, called the 5’ end and the 3’ end, and DNA sequences are generally written in the 5’ → 3’ direction [18].

DNA molecules do not exist as single strands. Instead, two DNA strands wrap around each other to form a double-helical structure, with the strands running in opposite directions (the 3’ end of one strand is next to the 5’ end of the other). Within the double helix, nucleotides from one strand are “paired” with nucleotides on the opposite strand via hydrogen bonding. Crucially, this pairing is specific: A always pairs with T, and C always pairs with G. This provides a mechanism for DNA replication: the two strands can be separated, and the complementary bases can be filled in on either strand to form two new, identical double helices.

RNA is chemically similar to DNA, and also consists of four nucleotide basis: Adenine (A), Cytosine (C), Guanine (G), and Uracil (U). In RNA, Uracil pairs with Adenine instead of Thymine. Because of RNA’s chemical similarity to DNA, DNA can be used as a “template” for the synthesis of RNA molecules in a process called transcription. During transcription, specialized proteins unwind a portion of the DNA double helix, and an RNA strand is synthesized with bases that are complementary to one of the exposed DNA strands. This process allows for portions of an organism’s DNA to be copied and transported throughout the cell, and it is constantly happening in all living organisms. Crucially, only small segments of DNA are transcribed into each molecule of RNA. Individu-
ally transcribed segments of DNA are called “genes,” and often encode a single protein.

After a gene has been transcribed into RNA, the RNA is usually further decoded into a protein sequence in a process called translation. Translation occurs in the ribosome, which is a molecular machine composed of RNA and proteins. For the purposes of this thesis, it is sufficient to know that each three “letters” of an RNA strand are translated by the ribosome into a specific amino acid on a growing protein chain. The correspondence between three-letter RNA segments (called codons) and amino acids is universal among all organisms, and because there are $4^3 = 64$ codons and only 20 amino acids, the code is redundant. The ribosome reads the codons in the 5’ → 3’ direction, and elongates the protein chain from the N terminus to the C terminus.

2.2 Protein Structure and Function

Thus far, we have reviewed the basic biology and chemistry of proteins. We are now ready to describe the three-dimensional structure of proteins, and how those 3D structures give rise to diverse functions. We have already mentioned the double-helical structure of DNA, which is the same regardless of the sequence of nucleotides making up the DNA helix. Unlike DNA, different protein sequences give rise to wildly different protein structures. In this section, we will begin by describing the geometric degrees of freedom present in proteins. We will then describe the different types of conformations adopted by natural proteins, and the biophysical forces that drive them to adopt those conformations.

2.2.1 Geometric Degrees of Freedom

In the previous section we described the chemical structure of proteins, which consists of amino acids joined by peptide bonds. Each amino acid residue in a protein chain consists of a central $\text{C}\alpha$ atom, which is bonded to a carbonyl carbon (so named because it is a member of a carbonyl C=O group), and an amide nitrogen (which is part of an amide group). Finally, the peptide bond connects the carbonyl carbon of one amino acid residue with the amide nitrogen of the subsequent residue. The sequence of bonded amide nitrogens, $\text{C}\alpha$ atoms, and carbonyl carbons (along with the carbonyl oxygen and the various hydrogens bonded to these atoms) forms the protein backbone. The full conformation of
a protein is given by the conformation of the backbone, and the conformations of the individual side chains.

What geometric degrees of freedom are present in the protein backbone? In a single amino acid residue, there are two rotatable covalent bonds: the N-Cα bond, and the Cα-C bond. While the peptide bond itself is a single bond, it has some characteristics of a double bond. As a consequence, the peptide bond does not rotate, and the bonded carbonyl carbon, amide nitrogen, and their corresponding Cα’s are fixed in the same 2D plane.
(referred to as the “amide plane”). These geometric degrees of freedom and constraints are illustrated in Figure 2.4A. Therefore there are two degrees of freedom per amino acid residue: the angle of rotation about the N-Cα bond, and the angle of rotation about the Cα-C bond. These angles are denoted as $\phi$ and $\psi$, respectively [18].

Although the peptide bond cannot rotate freely and violate the coplanarity of the atoms in the amide plane, it can adopt two distinct conformations which maintain the amide plane. These are referred to as the *cis* conformation, in which the Cα atoms from neighboring residues are on the same side of the peptide bond, and the *trans* conformation, in which neighboring Cα’s are on opposite sides of the peptide bond. These two distinct conformations are depicted in Figure 2.4B. To describe whether a peptide bond has a cis or trans conformation, we can introduce a third angle $\omega$, which has a value of 0 degrees in the cis conformation, and 180 degrees in the trans conformation. In practice, real peptide bonds are almost always in the trans conformation.

Therefore we can fully describe the geometry of a protein backbone using a series of angles: a $\phi$ and $\psi$ angle for each amino acid residue, and a $\omega$ angle for each peptide bond. The $\phi$ and $\psi$ angles can vary continuously, while the $\omega$ angle is either 0 or 180 degrees.

In theory, the $\phi$ and $\psi$ angles are free to rotate arbitrarily. However, some combinations of $\phi$ and $\psi$ angles bring non-bonded atoms too close together, resulting in strong electrostatic repulsions. If we plot the set feasible $\phi$ and $\psi$ angles actually observed in nature, we get a so-called Ramachandran diagram. An example Ramachandran diagram is given in Figure 2.5. Different portions of the Ramachandran diagram correspond to different types of local protein structures, which we will discuss momentarily.

### 2.2.2 DRIVERS OF PROTEIN FOLDING

The flexibility of the $\psi$, $\phi$, and $\omega$ angles allows for proteins to occupy an extremely vast conformational space, even when restricted to allowed regions of the Ramachandran diagram. What causes proteins to fold into one conformation but not others? The thermodynamic hypothesis, which has broad support from theory and experiments, states that proteins fold spontaneously to arrive at a conformation that minimizes free energy. This lowest-energy conformation is called the “native structure,” and it is fully determined by the protein’s sequence.
Free energy is an important concept in thermodynamics. Processes that decrease the free energy of a system are said to be thermodynamically favorable, and can therefore occur spontaneously. The formation of interactions like salt bridges and hydrogen bonds results in a decrease in free energy, while the breaking of these bonds increases free energy. Changes in a system that increase entropy (or disorder) are also thermodynamically favorable. This means that systems spontaneously become more disordered on average.

The concept of free energy is important for understanding the hydrophobic effect, which is a crucial driver of protein folding. When hydrophobic (i.e., nonpolar) molecules are introduced to water, they disrupt strong hydrogen bonds between water molecules and replace them with far weaker interactions, resulting in an increase in free energy. In addition, water molecules near the hydrophobic molecule become more ordered than normal, forming a crystalline pattern that maximizes hydrogen bonding with other water molecules and minimizes interaction with the hydrophobic surface. This decreases the entropy of the system relative to normal water, further increasing free energy. Therefore, it is thermodynamically favorable for hydrophobic compounds to minimize their interactions with water,
since it allows the water molecules to form more hydrogen bonds with one another, and to relax into a more disordered state. For this reason, hydrophobic compounds like oil segregate themselves when exposed to water in order to minimize contact [18].

Except for proteins that are embedded in cell membranes, proteins are always surrounded by water, so interactions between amino acid residues and water are key to protein folding. The dynamics of protein folding are driven primarily by the hydrophobic effect: proteins fold so that their hydrophobic side chains are packed together in the core of the protein, which minimizes free energy. Optimizing the packing of hydrophobic side chains without disrupting other favorable interactions is an extremely complex process, and results in a wide array of different protein structures.

As an aside, the notion that the structure of a protein is fully determined by its amino acid sequence (also referred to as its primary sequence) will become crucial later in this thesis, so it is worth exploring in more detail here. This fact was first demonstrated by the pioneering experiments of Christen Anfinsen in 1961 [2]. At a high level, Anfinsen purified a protein called ribonuclease-A, and fully unfolded it through the introduction of various chemical agents. When the protein was unfolded, it was no longer able to perform its function. However, when the chemical agents were removed, the protein refolded and regained its function, demonstrating that protein folding can occur spontaneously in a test tube. Other clever experiments demonstrated that the chemical agents truly did unfold the protein, rather than simply inhibiting its function.

While we have established that proteins fold to minimize free energy, it should be noted that it is impossible for proteins to search every possible conformational state to find the one with the lowest energy. Even if we assume a small number of possible $\phi$ and $\psi$ angles per residue and the ability to explore billions of states per second, it would still take millenia to explore all of the states for a reasonably-sized protein (on the order of 100 residues). This issue was first articulated by Cyrus Levinthal, and is known as Levinthal’s Paradox [46]. Therefore proteins must fold by descending a relatively smooth “free-energy energy landscape” in which smaller segments of the protein can fold locally before combining into a global geometry. The hierarchy from local to global structure in proteins is explored in the next two sections.
2.2.3 Secondary Structure

Each residue in the protein backbone contains a number of polar groups: the N-H group containing the amide nitrogen is a strong dipole, since nitrogen is more electronegative than hydrogen. The carbonyl (C=O) group is also polar, with the oxygen having a partial negative charge. When exposed to water these groups will form hydrogen bonds, with the carbonyl oxygen acting as a hydrogen bond acceptor, and the amide nitrogen acting as a donor.

When a protein folds, some of the backbone must necessarily be buried inside the core of the protein, precluding it from forming hydrogen bonds with water. If these hydrogen bonds were not replaced somehow, this would created a large energetic penalty associated with protein folding, making folding less thermodynamically favorable. To solve this problem, the backbone can fold to form hydrogen bonds with other sections of the backbone when it cannot hydrogen bond with water. To accomplish this, that backbone forms local structures called alpha-helices and beta-sheets. The formation of alpha helices and beta sheets by various sections of a protein is called secondary structure [18].

Alpha helices are a ubiquitous secondary structural element of proteins. In an alpha helix, the protein backbone twists into a right-handed helical structure, with the correct periodicity such that the C=O group of one residue can hydrogen bond with the NH group of the residue four positions later. Therefore, the formation of an alpha helix allows for all of the polar backbone groups (aside from three on each end) to form hydrogen bonds. An alpha helix is illustrated in Figure 2.6A. In an alpha helix, the side chains point outward away from the center, and all $\phi$ and $\psi$ angles for residues in the helix fall into a specific region of the Ramachandran diagram. (Figure 2.5).

The other major secondary structure element is the beta sheet. Beta sheets are composed of multiple beta strands, in which the backbone has an extended, linear conformation. When two beta strands line up next to each other, the C=O groups and NH groups can hydrogen bond with these same groups on the opposite strand. Beta sheets can consist of anywhere from two to dozens of aligned beta strands. Except for the strands on the ends (which have half of their polar groups exposed to the water) this arrangement allows for all of the polar backbone groups to hydrogen bond. A beta sheet is pictured in Figure 2.6B. In a beta sheet, the side chains point out of the top and bottom of the sheet.
Like in alpha helices, residues in beta sheets all generally occupy the same region of the Ramachandran diagram. (Figure 2.5).

Some regions of proteins are in neither alpha helices or beta sheets. These regions are called “loops,” and they are generally small segments connecting regions with helical or sheet secondary structure. Loop regions are usually exposed to water, so they do not have to create a special geometry in order to satisfy their backbone hydrogen bond donors and acceptors. The $\phi$ and $\psi$ angles of loop regions do not occupy specific regions of the Ra-
machandran diagram, since they do not have a regular, repeating structure. A loop region is highlighted in Figure 2.6C.

2.2.4 Tertiary Structure

Based on the previous sections, we know that proteins fold to minimize free energy by forming a hydrophobic core, and that polar groups on the protein backbone hydrogen bond with one another to form secondary structural elements. The tertiary structure of a protein describes how various sheets and helices pack against one another to bury hydrophobic side chains and expose hydrophillic ones.

There are a diverse set of possible tertiary structures for proteins, but many similar structural “motifs” are frequently observed across nature. For example, alpha helices often form with hydrophobic side chains on one side of the helix, and hydrophillic residues on the other side. Several such helices can come together to form a bundle with the the hydrophobic residues buried on the inside. Such a bundle is illustrated in figure 2.7A. Similarly, beta strands can arrange themselves into a circular beta sheet, with the last strand and the first strand pairing together. This tube-like structure is called a “beta barrel,” and hydrophobic side chains are located at the interior of the barrel, while hydrophillic side chains are on the exterior. An example of the beta barrel motif is shown in Figure 2.7B. These two motifs, as well as many others, are examples of the diverse tertiary structures found in proteins. Proteins can also exhibit quaternary structure, in which multiple different proteins form complexes together. Although many proteins do have quaternary structure, this thesis is mostly focused on the prediction and analysis of secondary and tertiary structure [18].

The tertiary structures of proteins can often be separated into distinct units called “domains.” Domains are typically 50-200 residues in length, and have their own hydrophobic cores. Different domains of the same protein can carry out different functions, and the same modular domains can be found in multiple different proteins. An example of a protein with multiple domains is given in figure 2.7C.
2.2.5 The Structure-Function Relationship

We have already alluded to the fact that proteins perform a myriad of functions within living cells. These functions include replicating DNA, moving muscles, transporting nutrients across cell membranes, metabolizing sugars, sensing light, conducting nerve impulses, and many, many more. In this section we will highlight one example of how proteins’ specific structural features allow them to accomplish these functions.

Ion channels are proteins that are critical to the function of nervous systems, among other things. In order for nerve cells to fire, it is necessary for potassium ($K^+$) and sodium ($Na^+$) ions to flow in and out of the cells at specific times. This means that the channels
which conduct these ions must be specific — a potassium channel must only allow potassium into the cell, and not other atoms. The potassium channel must also transport ions across the cell membrane very quickly. Amazingly, organisms have evolved a protein which meets exactly these requirements.

Figure 2.8 shows the structure of a potassium channel that allows potassium ions (and only potassium ions) to leak out of the cell and into the extracellular space. It contains four individual polypeptide chains, each of which is a bundle of alpha helices, and the four chains surround a central pore. Like all transmembrane proteins, this channel has almost all hydrophobic residues in the segment that is inserted into the membrane. On the intracellular side, the channel has negatively charged residues that attract positively charged K^+ ions into the pore. The pore is lined with a special “selectivity filter” composed of carbonyl oxygens from loop regions in the peptide backbone. Before entering the filter, the potassium ion has hydrogen bonds with several surrounding water molecules, but the filter is too small for both the ion and its associated waters. However, the geometry of the negatively charged oxygens in the filter is exactly the right size so that K^+ ions can lose their hydrogen bonds with waters and interact with the oxygens instead, without paying an energetic penalty. In contrast, smaller sodium ions would not be able to interact as favorably with the filter, so they would pay an energetic penalty for giving up their bonds with water. This precisely-tuned structure gives the protein a specific functional characteristic: its specificity for potassium [18].

2.3 Experimental Determination of Protein Structures

Given the applications enabled by the knowledge of protein structures, determining the structures of relevant proteins is of great scientific interest. Since the focus of this thesis is on the computational prediction of protein structures, it is not necessary to fully understand the intricacies of the experimental methods used to determine protein structures. However, it will be useful to provide a brief overview of some of the most common methods in this section. After protein structures are determined using these methods, they are typically deposited in the Protein Data Bank (PDB). Each protein structure in the PDB has a unique four-character identification code, and we have used these PDB codes to refer to specific protein structures throughout this thesis.
2.3.1 X-Ray Crystallography

X-ray crystallography is the oldest, highest-resolution, and most widely-used method for determining protein structures. It was used by John Kendrew to solve the first ever protein structure (the structure of myoglobin) in the 1950s. A crystal is a chemical structure where the atoms form a regularly-repeating pattern. For example, water molecules in an ice crystal make hydrogen bonds to form sheets of stacked hexagons. When a crystal is illuminated by X-rays, the X-rays diffract off of the regularly-repeating structure of the crystal, and form a characteristic diffraction pattern. This diffraction pattern can be recorded by a detector and used to reconstruct the 3D structure of the illuminated crystal \([18]\).

In order to solve the structure of proteins using X-ray crystallography, the protein must first be crystalized. This means that many copies of the same protein must pack against each other in a regularly repeating pattern, forming a crystal. To grow a crystal of a protein, the protein must be synthesized in a solution that is extremely pure with very high concentration. The solution containing the protein is then brought to a state of supersaturation through the addition of reagents that reduce the solvent’s ability to dissolve the protein. This causes some of the proteins to come out of solution and pack together into a growing crystal. Growing a crystal of a protein is often a laborious process, and requires a lot of trial-and-error before the exact conditions required for satisfactory crystalization are
found [25].

Once a crystal is grown, it can be illuminated by X-rays to form a diffraction pattern. This diffraction pattern is, roughly speaking, the Fourier transform of the density of electrons inside the crystal. By performing an inverse Fourier transform, it is possible to reconstruct the pattern of electron density inside the crystal [38].

2.3.2 Nuclear Magnetic Resonance Spectroscopy

Nuclear Magnetic Resonance (NMR) Spectroscopy is another widely-used technique for determining protein structures. At a high level, NMR spectroscopy utilizes the fact that atomic nuclei absorb and emit specific wavelengths of radiation when exposed to a strong magnetic field, and that those wavelengths depend on the chemical environment of the nucleus.

To analyze a protein structure using NMR, many molecules of the protein are suspended in solution, exposed to a high magnetic field, and irradiated with radio waves. Following the pulse of radio waves, the atomic nuclei in the proteins will relax into lower-energy states, emitting radio waves at specific frequencies. By analysing the frequencies of radio waves emitted from the sample, it is possible to match certain peaks in frequency with specific nuclei in the protein, since the frequency of emitted radiation is dependent on the chemical environment of the nucleus [8].

The emission of radio waves by magnetized nuclei is a result of their nuclear spin. When nuclei are close to each other, their spins are capable of influencing one another. Therefore, the magnitudes of frequency peaks corresponding to nearby nuclei are correlated in NMR experiments. By running many NMR experiments with varied parameters (like the amount of time between successive radio pulses) it is possible to detect correlations between specific peaks in the NMR spectrum. The allows for the identification of nuclei that are nearby in the folded structure, even if they are far apart in the linear amino acid sequence. Such information can be used to reconstruct the tertiary structure of the protein [8].

NMR has the advantage that it does not require proteins to be crystalized. However, NMR is less effective for large proteins, and typically gives lower-resolution structures than X-ray crystallography.
2.3.3 Cryo-Electron Microscopy

Cryo-Electron Microscopy (Cryo-EM) is a relatively recent technique for determining protein structures. It relies on the flash-freezing of single protein molecules, followed by imaging with an electron microscope. While Cryo-EM structures are typically lower in resolution than those produced with X-ray crystallography, their accuracy is improving due to advancing technology, and Cryo-EM structures do not require crystalization.

The basic principle of electron microscopy is to “illuminate” a sample under in a vacuum with a beam of electrons, and then measure the scattering of those electrons with specialized detectors. This is conceptually similar to the function of a light microscope, except the wavelengths of electrons (in particular, their \textit{de Broglie} wavelengths) are much smaller than those of visible light, which allows for much greater imaging resolution, including the imaging of single atoms \cite{27}.

Unfortunately, when imaging proteins with an electron microscope, the electron radiation causes significant damage to the protein by breaking covalent bonds, thereby destroying the structure being imaged. However, by freezing specimens of the protein in a thin layer of ice, it is possible reduce this structural damage while still obtaining high-resolution images. When determining the structure of a protein via Cryo-EM, many copies of the protein are frozen within the ice and imaged, which gives many different 2D-views of the protein. These images can then be recombined into a 3D model of the protein \cite{27}. While Cryo-EM does not require crystalization, the protein must still be isolated at very high concentrations and levels of purity, and determining the correct conditions for satisfactory freezing can be difficult.

2.4 Evolutionary Relationships Between Proteins

We will conclude this chapter by discussing evolutionary relationships between proteins. Since an organism’s protein sequences are encoded directly into its genome, the rise of fast, cheap DNA sequencing technology has allowed for the large-scale comparison of evolutionary related (or \textit{homologous}) protein sequences across many different species. Such analyses have revealed that many protein sequences, structures, and functions are conserved across distantly related branches of the tree of life.

As an example, consider the protein hemoglobin, which is used by humans (and almost
all other vertebrates) to transport oxygen inside red blood cells. Given the importance of this function, it is unsurprising that the human hemoglobin protein is highly similar to the hemoglobin used by related species in terms of both protein sequence and structure [18].

Given the qualitative similarity between proteins from related organisms, how can we quantify the degree of relatedness between similar protein sequences? One simple notion of relatedness between proteins is sequence identity. To compute sequence identity, two sequences are lined up, and the fraction of positions at which the amino acids are identical is calculated. While this metric is intuitively appealing, it is complicated by the fact that related proteins often have different lengths due to mutations that insert or delete segments of an organism’s DNA. When homologous proteins have different lengths, it is unclear how to line them up and calculate their identity to one another. This is known as the problem of finding a sequence alignment (i.e., an optimal correspondence between the positions of two related protein structures), and there are many algorithms to solve this problem. We will briefly describe one approach to give a flavor of how these algorithms work.

The Needleman–Wunsch algorithm is a basic approach for aligning two protein sequences [30]. The basic idea behind the algorithm is to define a score that reflects the quality of an alignment, and then to optimize that score using dynamic programming. In more detail, suppose we have two sequences \( A = a_1, a_2, \ldots, a_n \) and \( B = b_1, b_2, \ldots, b_m \), which we would like to align. Consider a candidate alignment \( A' = a'_1, \ldots, a'_\ell \) and \( B' = b'_1, \ldots, b'_\ell \) with \( \ell \geq \max(n, m) \). Here, \( A' \) and \( B' \) consist of the sequences \( A \) and \( B \) with interspersed “gap” characters (denoted as \( _- \)). To define the score optimized by the Needleman–Wunsch algorithm, we first need to specify a function \( G(k) \) that penalizes either sequence for having a gap of length \( k \), as well as a function \( s(a, b) \) which defines the penalty for aligning two non-gap characters \( a \) and \( b \). This penalty may simply be set to \(+1\) for mismatching amino acids and \(0\) for the same amino acids, or a more complex scoring function can be used that takes chemical similarities between amino acids into account. Given these functions, the alignment score is computed as follows:

\[
S(A', B') = \sum_{i=1}^{\ell} \begin{cases} 
  s(a'_i, b'_i) & a'_i \neq _, b'_i \neq _- \\
  G(\min \{ j : a'_{i-j} \neq _- \}) - G(\min \{ j : a'_{i-j} \neq _- \} - 1) & a'_i = _- \\
  G(\min \{ j : b'_{i-j} \neq _- \}) - G(\min \{ j : b'_{i-j} \neq _- \} - 1) & b'_i = _-
\end{cases}
\]
Using a dynamic programming approach, the Needleman–Wunsch algorithm computes the optimal alignment in $O(mn)$ time. In practice, one often wants to align a large number of sequences, so a worst-case quadratic time complexity is not acceptable. In these cases, more efficient heuristic algorithms like BLAST are often used [24]. While the preceding algorithm only aligns a single pair of sequences, it is also possible to jointly align many different homologous sequences into a *multiple sequence alignment*.

Now that we have covered the notion of a sequence alignment, it is possible to precisely quantify the sequence identity between related protein sequences, even if they have different lengths. Returning to the example of hemoglobin, we find that human hemoglobin has very high sequence identity with related organisms (98.6% for orangutan, 86.6% for mouse, and 81.7% for zebra), indicating a recent shared evolutionary origin. More distantly related “globin proteins,” such as those used in invertebrates and plants, have much lower sequence identities in the range of 10-20%. However, these globin proteins still have very similar 3D folds and biological functions to human hemoglobin. This highlights an important point about protein evolution: distantly related proteins that have quite dissimilar sequences can still share the same structure and function. This indicates that, in most regions of the protein, there are many mutations that can be tolerated without seriously damaging the protein’s function. However, certain residues (like those that are key for binding other molecules and catalysing chemical reactions) are extremely sensitive to mutations, and these residues tend to be conserved in related proteins even if other parts of the sequences are dissimilar [18].
Protein Structure Prediction: A Grand Challenge

In the previous chapter, we introduced the world of proteins, and discussed how they fold, function, and evolve. Throughout our discussion, we emphasized how the three-dimensional structure of a protein is closely related to its function, which makes an understanding of protein structure critical to a mechanistic understanding of biology, as well as applications like drug design. We also emphasized how protein structures are extremely diverse and complex, and therefore require time- and labor-intensive experiments to determine. However, according to the thermodynamic hypothesis, protein structures are fully determined by their amino acid sequences, and fold because of the thermodynamic push to bury their hydrophobic side chains. Therefore, if all of the information needed to determine a protein structure is present in its sequence, why can’t we simply predict the structure from the sequence, thus bypassing the need for costly experimental techniques? This is the problem of protein structure prediction, and it has been considered a grand challenge in molecular and computational biology for over 50 years. In this chapter we will cover the various approaches that have been used to make progress on this problem. This survey of approaches will culminate in a description of AlphaFold, a recent breakthrough structure prediction
method that has been described as “solving” the problem. We will conclude by discussing the aspects of the problem that remain unsolved, and how the research in this thesis contributes to those unsolved problems.

3.1 Preliminaries and Metrics

Before jumping into an exploration of methods for protein structure prediction, it will be helpful to cover a few preliminary concepts. At a high level, a protein structure prediction algorithm takes in a sequence of amino acids, performs a computation, and returns a 3D protein structure. This protein structure can be represented in a number of ways, including a set of \((\phi, \psi, \omega)\) angles or a set of 3D coordinates for each atom. Another useful representation is a C\(\beta\) distance matrix. This representation takes the form of an \(N \times N\) matrix (where \(N\) is the number of residues in the protein), where the \((i, j)\) entry is the distance between the C\(\beta\) atom of residue \(i\) and the C\(\beta\) atom of residue \(j\). If either of the residues is a glycine, the C\(\alpha\) atom is used instead. It is most common to return a set of 3D coordinates, although the exact translation and rotation of these coordinates is of course arbitrary.

After receiving a set of atomic coordinates from a protein structure prediction algorithm, it is often desirable to compare those coordinates with experimentally determined coordinates in order to assess the accuracy of the algorithm. How should we measure the “closeness” of a set of predicted coordinates to the native structure? Computational biologists have devised a number of different metrics to solve this problem, and those metrics are summarized in the following section.

The Root-Mean-Square-Deviation (RMSD) is the most basic metric of similarity between two protein structures. In general, the RMSD can be computed between two sets of points \(X = x_1, \ldots, x_n\) and \(Y = y_1, \ldots, y_n\), where we know there is a direct correspondence between \(x_i\) and \(y_i\) (in the case of a protein, \(x_i\) and \(y_i\) are the predicted and native positions of the \(i\)th atom in the protein’s molecular structure). The RMSD is computed as follows:

\[
\text{RMSD}(X, Y) = \min_{S \in \mathcal{S}} \sqrt{\frac{1}{n} \sum_{i=1}^{n} ||S(x_i) - y_i||^2}
\]

Where \(\mathcal{S}\) is the set of all rigid translations and rotations in \(\mathbb{R}^3\). While this optimization
problem may look intimidating, it can be solved very efficiently using the Kabsch algorithm, which takes advantage of the Singular Value Decomposition. When comparing protein structures, it is common to examine the RMSD between the Cα atoms only to assess the accuracy of the backbone modelling. For protein structures, RMSDs are usually measured in Angstroms (Å). One angstrom is equivalent to $10^{-10}$ meters.

The main drawback of RMSD is that it can be very large for protein models that are actually quite close to the native structure. For instance, if a single structural element is oriented incorrectly relative to the rest of the protein, but all of the local structure is modelled perfectly, the RMSD can still be very high. Also, predictions of larger proteins naturally have larger RMSDs, which does not necessarily reflect the quality of the underlying prediction. These issues have motivated the development of other metrics, including the TM Score, GDT_TS, and LDDT Score.

The TM Score is a commonly-used metric of protein model accuracy. It ranges from 0 to 1, with 0 meaning that a prediction has essentially no similarity to the native structure and 1 indicating a perfect prediction [45]. The TM Score objective function is given by the following formula:

$$\text{TM Score} = \max_{S \in S, I \subseteq \{n\}} \frac{1}{n} \sum_{i \in I} \frac{1}{1 + \|S(x_i) - y_i\|^2 / d_0^2}$$

where $d_0$ is a fixed constant. Because the TM Score is only calculated over a subset $I$ of the residues, it is able to “focus” on the regions that are correctly modelled. In this sense, the TM Score can be thought of as roughly corresponding to the highest possible fraction of residues that can be aligned with reasonable accuracy. This means that, unlike RMSD, TM Score is able to indicate that part of a structure is correct even if other parts are incorrect. The optimization problem in the TM Score objective does not admit a simple analytical solution, so it is approximated with an iterative heuristic algorithm. Also note that the TM Score only considers Cα atoms.

GDT_TS is another metric that is conceptually similar to TM Score. For a given “cutoff distance” $d$, the GDT_TS for that cutoff distance is the maximum fraction of predicted Cα atoms that can be superimposed on the native structure such that their distances to the native Cα atoms are all less than or equal to $d$ [44]. For low values of $d$ the GDT_TS will be close to 0, while the GDT_TS approaches 1 as $d$ goes in infinity. Generally, the fi-
nal GDT_TS is calculated by averaging the individual GDT_TS scores for $d = 1, 2, 4$ and $8 \, \text{Å}$.

Finally, the LDDT score is yet another metric for assessing the accuracy of predicted protein structures. To motivate the LDDT score, consider a protein with two domains, and suppose a structure prediction algorithm models each individual domain correctly, but the relative positions of the domains are incorrect. The RMSD for the predicted structure will be extremely poor, since no good superposition of the full structures exists. GDT_TS and TM Score will be more forgiving, and will reward the algorithm for predicting one of the two domains correctly. However, these scores will not reflect that both domains are predicted correctly, and only a few residues in the loop region connecting the domains are incorrect. To resolve this issue, proteins are often segmented into individual domains before being scored using GDT_TS and TM Score, but this segmentation creates an additional step that sometimes requires manual intervention. In order to resolve this issue, the LDDT score assesses the correctness of the local environment of each residue.

As before, let $x_1, \ldots, x_n$ by the coordinates of the prediction, and $y_1, \ldots, y_n$ be the coordinates of the native structure. The LDDT score is computed with respect to a single atom $i$ in the native structure. The neighbors of $i$ defined by $N(i) = \{ j : ||y_j - y_i|| < R_0 \}$, where $R_0$ is a fixed cutoff radius. The LDDT Score for atom $i$ is defined as:

$$
\text{LDDT} = \frac{1}{|N(i)|} \sum_{j \in N(i)} 1 \{ (||y_j - y_i|| - ||x_j - x_i||) < \delta \}
$$

Where $\delta$ is a fixed threshold. Intuitively, this metric determines the fraction of local interatomic distances in the native structure that are “preserved” in the prediction. The final LDDT score is computed by averaging this metric over all atoms in the native structure. Alternatively, it can be computed with respect to just the C$\alpha$ atoms, in which case it is called C$\alpha$-LDDT. Based on the formula above, the LDDT must be between 0 and 1 [23].

The development of these various metrics has largely been driven by the Critical Assessment of protein Structure Prediction (CASP). CASP is a biannual competition in which entrants are challenged to predict protein structures which have been solved, but whose ground-truth structures have not yet been made public. The first CASP was held in 1994 and, most recently, CASP14 occurred in 2020. The upcoming CASP15 is scheduled to be-
gin this May.

Now that we have covered metrics for assessing the accuracy of protein structure predictions, we are ready to explore some of the methods that have been used to make progress on this important problem.

3.2 Physics-Based Methods

In some sense, there is a natural algorithm for converting an amino acid sequence into a folded protein structure – namely, the algorithm given by running the laws of physics forward in time. Unfortunately, the exact simulation of large molecules using quantum mechanics is far outside the capabilities of modern computers, and will be for the foreseeable future. Therefore, techniques that aim to directly simulate the physics of protein folding must use approximations to ensure computational tractability, while maintaining enough detail to achieve accurate results. Different approaches lie on different sides of this tradeoff – from molecular dynamics methods that attempt to model atomic interactions from first principles, to methods that optimize empirically-derived potential functions designed specifically for protein folding.

3.2.1 Molecular Dynamics

Molecular dynamics simulations aim to model chemical systems at the level of individual atoms by simulating Newton’s laws of motion. At a high level, the state of a molecular system can be described by the positions of the atoms $\vec{x}_1, \ldots, \vec{x}_n$, as well as their momenta $\vec{p}_1, \ldots, \vec{p}_n$. If there is a known potential energy function $U(\vec{x}_1, \ldots, \vec{x}_n)$, then Newton’s laws perfectly describe the evolution of the system:

$$\frac{\partial}{\partial t} \vec{x}_i = \vec{p}_i / m_i$$

$$\frac{\partial}{\partial t} \vec{p}_i = - \frac{\partial}{\partial x_i} U(\vec{x}_1, \ldots, \vec{x}_n)$$

In principle, numerically integrating these equations of motion should allow for simulation of physical systems, as long as the length scales involved are large enough that quantum mechanical effects do not play a large role. In practice two important problems arise:
how to accurately model the potential energy function $U$, and how to precisely integrate the equations of motion in a way that is computationally feasible.

Modelling the energy function $U$ is a complex topic, but it ultimately boils down to making a series of computationally feasible and empirically accurate approximations. Interactions between charged particles are modelled using Coulomb’s inverse square law, the distances and angles between bonded atoms are modelled as harmonic springs, etc. The parameters for these approximations are generally derived by fitting to experimental data. These approximate energy functions are called force fields [21].

In order to accurately simulate molecular systems, the equations of motion must be integrated with an extremely short time step. This generates an extremely high demand for computational resources, especially since systems of interest can contain hundreds of thousands of interacting atoms. This issue has been addressed by developing highly efficient software for molecular dynamics simulations, as well as specialized hardware accelerators. Currently, the Anton 3 molecular dynamics supercomputer (developed by D.E. Shaw Research) can simulate a system of one million atoms for about 100 microseconds of simulated time per day of computation [36]. For reference, proteins can take anywhere from several microseconds to several seconds to fold, depending on their size and structure [18].

How accurately can molecular dynamics simulations predict the structures of proteins? In 2011, researchers from D.E. Shaw Research reported the use of molecular dynamics simulations to replicate the folding of 12 especially fast-folding proteins, all of which has fewer than 100 residues. These structures were highly accurate, with most achieving RMSDs lower than 2 Å[20]. Such results are a promising sign that molecular dynamics force fields are accurate enough to reproduce experimental results. However, subsequent research has generally not been able to extend these results to larger, slower-folding proteins. This may indicate that molecular mechanics force fields are not accurate enough to capture the energy landscape of complex proteins, or that we cannot simulate proteins long enough for them to overcome the potential energy barriers keeping them from reaching their folded state.
3.2.2  Rosetta

Given the computational intensity of molecular dynamics, which models all of the atoms in a protein as well as surrounding water molecules and ions, it is natural to wonder whether a protein-specific energy function could be developed to simulate the essential physics of proteins at lower computational cost.

The Rosetta energy function aims to do this by approximating the free energy landscape of protein conformations. Ideally, such an energy function should assign low energies to proteins in their native structures, and high energies to proteins in their unfolded states. The Rosetta energy function is based on both basic physical principles and heuristic terms derived from empirical data on protein structures. Like a molecular dynamics force field, Rosetta directly simulates the physics of Van der Waals forces using a Lennard-Jones potential, which is a U-shaped potential with its lowest point at the Van der Waals radius. Rosetta also uses Coulomb’s law, approximates the energies of hydrogen bonds, and adds a heuristic penalty for the exposure of hydrophobic side chains to the protein’s exterior. Additionally, Rosetta uses a potential function over the $\phi$ and $\psi$ angles of each amino acid residue. This potential term is based on the empirical distribution of backbone angles for each amino acid type that is observed in solved protein structures. A similar statistical potential is also used to score the conformations of the side chains [1].

The Rosetta energy function has been used successfully for many applications, including the design of novel proteins and the prediction of interface geometries between interacting proteins. In theory, the Rosetta energy function offers a method for protein structure prediction: simply find the protein conformation that returns the lowest energy score for a given amino acid sequence. Unfortunately, this approach runs into the issue of Levinthal’s paradox: the potential conformation space is extremely vast, and the Rosetta energy function is computationally expensive enough that extensive search remains difficult. In addition, gradient descent in the Rosetta energy function from a random initialization generally gets stuck in incorrect local energy minima. Therefore, it is necessary to use intelligent sampling algorithms to search the conformational space for low-energy structures.

The Rosetta protein structure prediction protocol proceeds using a Monte Carlo sampling approach. First, thousands of protein models are initialized in a fully extended conformation. Next, fragments of each protein backbone have their $\phi$ and $\psi$ angles randomly
replaced with angles from solved structures with similar local sequences (this is an example of homology modeling, which is an approach to structure prediction that looks for homologous proteins with solved structures). Following the Metropolis-Hastings algorithm, replacements are more likely to be accepted if they cause a decrease in the Rosetta energy. After a number of iterations the simulation terminates, resulting in several thousand sampled candidate structures. Note that, during this procedure, a simplified version of the Rosetta energy function is used that does not use any information on side chain geometries. This is because the backbone angles can easily be replaced with angles from homologous proteins, but it is much less clear how to determine the the side chain placements. Unfortunately, this “coarse-grained” energy function is significantly less accurate than the all-atom energy function [37].

After this initial sampling step, the candidate models are refined via gradient descent in the all-atom Rosetta energy function, and the lowest-energy models are chosen as predictions. Like folding via molecular dynamics simulations, this protocol has been successful for predicting the structures of small proteins less than 100 residues in length. However, it appears that the sampling of the conformational space is not exhaustive enough or the energy function is not accurate enough to enable the prediction of larger proteins with reasonable computational resources.

### 3.3 Coevolution: Biodiversity Reveals Structure

In the previous section, we discussed methods that aim to predict protein structures by simulating the physics of protein folding at various levels of abstraction. While these methods show promise for small, fast-folding proteins, they generally fail on larger proteins. The main reason for this failure seems to be that the landscape of possible protein conformations is too vast to be searched in a computationally efficient way, especially when more accurate and detailed energy functions are used. This motivates the question of whether any additional information aside from the amino acid sequence can be used to directly reveal information about the protein’s structure, thereby circumventing the difficult global search problem.

It turns out that there is a very elegant and effective way to reveal structural information about a protein by leveraging additional information: namely, by analysing patterns
of variation in evolutionarily related protein sequences. We will refer to methods that predict protein structures by analysing evolutionary variation as *coevolution methods*.

To understand how coevolution methods work, recall our discussion of evolutionary relationships between homologous proteins in different species. We noted that homologous protein sequences from related organisms have very similar structures despite having non-identical sequences, since many amino acid substitutions can be tolerated without significantly changing the folded protein structure. The central idea behind coevolution methods is that a mutation at one residue in a protein structure can exert new selective pressures on other residues, including residues that are far away in the linear protein sequence. This effect was first documented by Yanofsky et al. in 1964, who studied mutations in a bacterial protein responsible for synthesizing tryptophan. They noticed that, while an amino acid substitution at a specific residue made the protein non-functional, a second mutation at a position 36 residues away (in terms of the linear protein sequence) was able to restore the functionality of this mutant. This led them to hypothesize that these two positions were in close contact in the folded protein structure, since a chemical change in the second residue was able to compensate for the effects of a mutation in the first residue [43].

Subsequent research has revealed this to be a general principle of protein evolution: If two residues are in contact in a folded protein structure, mutations in the first position often induce a selective pressure for the second position to mutate. When analysing a family of related protein sequences, this effect makes it appear as though mutations at contacting positions are positively correlated (although technically the individual mutations occur at random, and the observed correlation is a result of natural selection) [29, 13, 3].

In order to turn such correlational patterns into robust predictions of protein structure, two ingredients are needed: a large number of related sequences, and a statistical model to distinguish true mutational correlations from background noise. Recall from our discussion of protein evolution that related proteins can be compiled into a multiple sequence alignment (MSA), which is essentially a 2D grid where each column is a sequence position, and each row is a homologous sequence from a different organism. Many algorithms have been developed to analyze MSAs and detect covariance patterns that indicate the presence of contacting residues in the folded protein structure. We will give a brief description of Markov Random Fields, which are one such class of algorithms.

Formally, let us consider an MSA with $M$ rows and $N$ columns, and let $a_{ij}$ denote the
entry in the $i$th row and $j$th column. The character $a_{ij}$ represents position $j$ in the protein sequence from organism $i$, and it may be either an amino acid identity or a gap symbol. Suppose that the first row $a_{1,\cdot}$ of the MSA contains the sequence we want to predict (the “target sequence”), and that it has no gaps (so $N$ is the length of the target sequence). The sequences in rows 2 through $M$ have been aligned to the target sequence using a procedure similar to the Needleman–Wunsch algorithm described earlier.

To model the sequences in the MSA statistically, we can assume that each sequence in the MSA is drawn independently from an unknown probability distribution:

$$a_i \overset{i.i.d.}{\sim} \mathcal{D}$$

The distribution $\mathcal{D}$ of protein sequences is imposed by the evolutionary constraints on the protein’s function, meaning that protein sequences with higher “fitness” have higher probability. The distribution $\mathcal{D}$ should model the distribution of amino acid types at each position, as well as the correlations between different positions in the sequence. A Markov Random Field parameterizes a family of distributions $\mathcal{D} = (\mathcal{E}, \Phi, \Psi)$ over sequences as follows:

$$\mathcal{E} \subseteq \{(i,j) : i < j, i,j \in [N]\}$$ is a set of undirected edges between correlated positions in the sequence.

$$\Phi_i = \left[ \exp \left( \mu_1^{(i)} \right), \ldots, \exp \left( \mu_k^{(i)} \right) \right]$$ is a set of probabilities (unnormalized) for the various amino acids at position $i$.

$$\Psi_{i,j} = \begin{bmatrix}
\exp \left( \nu_{1,1}^{(i,j)} \right) & \cdots & \exp \left( \nu_{1,k}^{(i,j)} \right) \\
\vdots & \ddots & \vdots \\
\exp \left( \nu_{k,1}^{(i,j)} \right) & \cdots & \exp \left( \nu_{k,k}^{(i,j)} \right)
\end{bmatrix}$$

is a set of unnormalized probabilities for each possible pair of amino acids at positions $i$ and $j$.  

37
The likelihood of the MRF for a sequence $x$ is given by

\[
\frac{1}{Z} \prod_{i=1}^{n} \Phi_i(x_i) \prod_{(i,j) \in E} \Psi_{i,j}(x_i, x_j)
\]

Where $Z$ is a normalizing constant that depends on the parameters. Optimization of this objective over of $(E, \Phi, \Psi)$ is computationally intractable, so in practice approximations are used, often with regularizers that prevent overfitting and promote sparsity in the graph $E$. After fitting, the edges in the graph $E$ indicate correlated positions, and the norms of the correlation matrices $\Psi_{i,j}$ indicate the strength of the coupling. Empirically, edges with strong couplings tend to predict contacts in the folded protein structures [3]. To visualize the output of an MRF, it is common to create an $N \times N$ matrix, with the $(i, j)$ entry corresponding to strength of the coupling between positions $i$ and $j$ in the MRF. Figure 3.1 shows such a “contact map,” overlaid against the ground-truth plot of contacts in the folded structure. To convert the predicted contacts into predicted atomic coordinates, it is common to run dynamical simulations with molecular dynamics or Rosetta, with added mechanical restraints to enforce evolutionarily predicted contacts.

Coevolution methods can perform well on contact prediction for large proteins, and do not suffer from the scaling issues associated with physics-based methods. Some complications include the filtering out of false contacts, including those that arise from proteins that form complexes with other molecules of the same protein, which results in mutational correlations between residues that do not touch in a single copy of the folded structure. The folding of large proteins using constrained physics-based simulation can also be computationally difficult, even with the constraints provided by evolutionary analysis. Finally, coevolution methods cannot perform well unless there are a large number of homologous sequences to the target protein with considerable diversity.

While this last point is sometimes a significant issue, the rapid rise of genome sequencing technology has caused the number known protein sequences to explode over the past two decades. Such an abundance of sequence data has contributed to the effectiveness of coevolution methods for protein structure prediction. As a result, coevolution-based methods has been the principle drivers of progress in protein structure prediction over the last decade. As noted by Fraizer et al., the success of coevolution methods is a remarkable ex-
ample of how earth’s biodiversity is an invaluable resource, which we as a species must preserve and protect [10].

3.4 Contributions from Deep Learning

While the rise of coevolution methods has led to improvements in protein structure prediction over the last decade, further improvements in the last five-or-so years have come from the analysis of coevolution data with techniques from deep learning.

At a high level, deep learning simply refers to the training of large neural network models, often with millions or billions of parameters. Deep learning methods have been extremely successful on a wide variety of problems, including image recognition, language translation, playing games like Go and Atari, the synthesis of realistic images and texts, and many others. Research in deep learning has accelerated dramatically over the last decade due to improvements in computational hardware, an abundance of digital datasets, accessible software libraries for developing deep models, and enthusiasm from many segments of industry and academia. While deep learning has been extremely successful in
many application areas, deep models have a high demand for computational resources and training data, and it is difficult to interpret the inner workings of deep models to understand what makes them so successful.

This thesis assumes a familiarity with the basic structure of neural network models. However, we will briefly review several important neural network architectures and computational primitives that have been important for protein structure prediction. The first artificial networks were inspired by the computations performed by neurons in the brain. Such networks (sometimes called “Multilayer Perceptrons”) have several layers, each of which consists of a linear transformation followed by a nonlinear “activation function.” Mathematically, this computation proceeds as follows:

\[ x \in \mathbb{R}^n \text{ is the network input} \]
\[ y \in \mathbb{R}^m \text{ is the network output} \]
\[ W_1, \ldots, W_\ell \text{ are weight matrices with appropriate dimensions} \]
\[ \sigma(\cdot) \text{ is a fixed, nonlinear activation function} \]
\[ y = W_\ell (\sigma (W_{\ell-1} \sigma (\ldots W_2 (\sigma (W_1 x)) \ldots))) \]

In an MLP, the parameters of the network are the weight matrices, and they are tuned to compute a desired function of the input \( x \). In the intervening years, neural networks have evolved to encompass an extremely wide variety of nonlinear functions \( f_\theta (\cdot) \), with the common features that they are specified by a parameter vector \( \theta \), and that the function output is differentiable with respect to the parameters. As mentioned earlier, modern models may have parameter vectors \( \theta \) with millions or billions of tunable elements. Neural networks are generally trained with a variant of Stochastic Gradient Descent, which proceeds as follows:

Let \( (x_1, y_1), \ldots (x_n, y_n) \) be a data set
Let \( \mathcal{L}(y, \hat{y}) \) be a loss function, and let \( \alpha \) be a learning rate.
\[ \theta_0 \leftarrow \text{RandomInit()} \]
for \( i \) in \( 1 \ldots T \):

40
\[ j \sim \text{Unif}(1, n) \]
\[ \theta_i \leftarrow \theta_{i-1} - \alpha \nabla_{\theta} \mathcal{L}(y_j, f_{\theta_{i-1}}(x_j)) \]

return(\(\theta_T\))

The loss landscapes created by deep neural networks contain many local minima, so convergence of this training procedure is not guaranteed. Nonetheless, SGD works well in practice for training neural networks, and the minima found by SGD tend to generalize well on unseen examples. Additional regularization measures, such as “dropout,” which randomly zeros out the activations of model neurons, are sometimes applied during training to prevent overfitting.

The specification of the parametric function \(f_{\theta}(\cdot)\) is called the design of a “neural network architecture”. Although it is something of an art form, designing a good architecture is extremely important for achieving good performance on deep learning tasks. We will give a brief overview of Convolutional Neural Networks and Transformers, which are two architectures that are important for protein structure prediction. These descriptions will omit a lot of technical details, but will capture the essential computations involved. For more detailed explanations see the referenced works.

Convolutional Neural Networks were designed for image recognition tasks, and are loosely modelled after the organization of a mammalian visual cortex \([17]\). A single layer of a convolutional neural network can be seen as mapping a “input image” \(X \in \mathbb{R}^{N \times N \times D}\) onto an “output image” \(Y \in \mathbb{R}^{N \times N \times L}\). Here, the third dimensions \(D\) and \(L\) can be seen as representing the number of “layers” in the images, which can represent (for example) the number of colors. The convolutional network computes this mapping by sweeping \(L\) different “filters” over the input image to produce each layer of the output image. Let \(F^{(\ell)} \in \mathbb{R}^{3 \times 3 \times D}\) be the \(\ell\)th filter (the filter can have any width, but 3 is a standard choice). The computation proceeds as follows:

\[
Y_{i,j,\ell} = \sigma \left( \sum_{i'=1}^1 \sum_{j'=1}^1 \sum_{d=1}^D F^{(\ell)}_{i'+2,j'+2,d} \cdot X_{i+i',j+j',d} \right)
\]

Intuitively, each location \((i, j)\) in the output image is created by taking the dot product of the filter and a small region surrounding \((i, j)\) in the input image. These filters can
learn to detect features like edges in the input image, and then identify them in the output image. In a CNN, many of these convolutional layers are stacked, and the parameters of the filters are learned through gradient descent. If a prediction needs to be made at the end of the network, the final output image can be collapsed into a vector or scalar output using a linear projection [17].

Transformers are another important class of neural network architectures that have received a lot of attention in recent years. While CNNs were designed for analysing images, Transformers were designed for analysing written language and performing tasks like machine translation [42]. In the previous section, we explained how convolutional layers can be interpreted as mapping an input image into an output image. Analogously, a transformer layer can be seen as mapping an “input sentence” $X \in \mathbb{R}^{N \times D}$ to an “output sentence” $Y \in \mathbb{R}^{N \times L}$. Here $N$ is the number of “words” in the sentence, and $D$ and $L$ represent the lengths of the vectors used to represent each word.

At the heart of a transformer layer is a computational primitive called “attention.” First, each word $x_i$ of the input sentence are transformed into three distinct representations: queries, keys, and values. This occurs via matrix multiplication as follows:

$$q_i = Qx_i, \quad v_i = Vx_i, \quad k_i = Kx_i$$

with $Q, K \in \mathbb{R}^{D \times h}$ and $V \in \mathbb{R}^{K \times L}$. All of these weight matrices are learned parameters of the model. These representations are then re-aggregated to form the new word representations $y_i$:

$$w_i = \text{softmax} \left( \frac{q_i^T k_1}{\sqrt{h}}, \ldots, \frac{q_i^T k_N}{\sqrt{h}} \right),$$

$$y_i = \sum_{j=1}^{N} w_{ij} v_j$$

Where “softmax” denotes a frequently used function for turning an arbitrary vector into a vector probabilities that sum to 1:

$$\text{softmax}(v) = \left( \frac{\exp(v_1)}{\sum_{i=1}^{d} \exp(v_i)}, \ldots, \frac{\exp(v_d)}{\sum_{i=1}^{d} \exp(v_i)} \right)$$
The fundamental idea behind attention is that it allows for the output representations $y_i$ to selectively integrate information from many input positions $x_i$ from throughout the sequence. The attention weight $w_{ij}$ indicates how much the output $y_i$ is “paying attention” to the vector $x_j$ in the input sequence. After this attention transformation, the vectors $y_i$ are typically passed through a single MLP layer before the process repeats again [42].

At the end of a stack of transformer layers, an output representation vector is returned for each word in the sentence. Call these representations $z_i \in \mathbb{R}^L$. These representations are generally used to compute some output value of interest, and the correctness of that output value is used to train the weights of the model via gradient descent.

One common scheme for training transformers is called Masked Language Modelling. In this task, a random subset of the words in the input sentence are replaced with a special “mask” symbol. If the word at position $i$ is masked out, then the final representation $z_i$ from that position is used to “fill in the blank” and predict the hidden word. It has been found that, when transformer models are trained to predict masked-out words, they learn very rich representation vectors for the words in the input sentence. This means that, for example, simple linear models that use the output vectors $z_i$ as inputs are able to predict complex properties of the input sentence without a lot of training data, since the transformer has learned to pack a lot of useful information into its output vectors. This is an example of a learning paradigm called self-supervision, in which a model is able to learn useful representations from unlabelled data [6].

Note that, for both the CNN and Transformer architectures described above, the same models weights work even as the input size $N$ varies. This is important for processing inputs like protein sequences, which do not have a fixed size.

Now that we have reviewed the basics of deep learning and some relevant architectures, we are ready to explore how these architectures have made an impact on protein structure prediction.

### 3.4.1 AlphaFold1

AlphaFold, which we will discuss in detail soon, is a breakthrough deep learning model which has been heralded as a “solution” to the protein structure prediction problem. AlphaFold’s breakthrough performance came during the CASP14 competition in 2020. How-
ever, the version of AlphaFold from CASP14 is actually AlphaFold2. AlphaFold1 was
developed by DeepMind for CASP13 in 2018, and was an impressive advance in protein
structure prediction in its own right. Because the research community often refers to Al-
phaFold2 as simply “AlphaFold,” this thesis will use “AlphaFold,” “AlphaFold2,” and
“AF2” interchangeably to refer to the AlphaFold model from 2020. When referring to the
model from 2018, we will always use the term “AlphaFold1.”

Earlier in this chapter, we discussed how Markov Random Fields and related methods
can be used to analyse MSAs and produce an $N \times N$ map, in which large values are in-
dicative of potential contacts in the folded protein structure. While this is undoubtedly
useful, some questions about the predicted contacts remain. Which contacts are real, and
which ones are noise? How large must the coupling strength be for a putative contact to
be trusted? Just how close together are the contacting residues? Is it possible to “fill in”
missing contacts that were not revealed by the MSA? These are complications that can
potentially be addressed by a learning algorithm. AlphaFold1 does this by feeding MSA-
derived coupling information into a deep neural network, and then training the network to
predict more precise structural information $[35]$.

More specifically, AlphaFold1 constructs an $N \times N \times D$ input “image” by stacking a
matrix of couplings from an MRF-like model, as well as horizontally and vertically tiled
copies of the target amino acid sequence. This “image” is then passed through a convo-
lutional neural network, resulting in a $N \times N \times L$ output image. The $(i, j)$ entry of the
output image is an $L$-dimensional vector of probabilities, where each entry is the predicted
probability that the distance between residues $i$ and $j$ falls into a specific interval. This
type of output is referred to as a “distogram,” and it represents a predicted distribution
over inter-residue distances. To generate atomic coordinates, simulation in a Rosetta-like
potential is performed, with an additional energy term penalizing deviations from the pre-
dicted distribution of distances.

AlphaFold1 was the top model at CASP13 by a significant margin. However, it was far
from perfect, and predicted less than half of the CASP13 targets with a TM Score higher
than .7. Thus, while AlphaFold1 was an important step forward, it would take until Al-
phaFold2 at CASP14 for truly breakthrough performance to emerge from deep learning
methods.
3.4.2 **Deep Language Models**

In parallel to work on supervised protein structure prediction, other efforts have been made to apply deep language models like Transformers to the vast amounts of unlabelled protein sequence data present in genomic databases. In particular, a number of works have trained Transformers using Masked Language Modelling to predict masked-out amino acid residues in protein sequences [34]. It is clear that such language models have been able to learn extremely informative representations of protein structure. One unanticipated application of protein language models is that the sequence attention weights (denoted $w_{ij}$ in the exposition above) can be readily used to predict contacts in folded protein structures, much like MSA-based couplings. Indeed, the strength of the attention weights between two residues is an even better predictor of protein contacts that statistical models like Markov Random Fields [33]. Contacting residues often mutate in tandem, so it makes sense that the model “pays attention” to geometrically close residues to predict the identities of masked-out amino acids. Still, it is impressive that Transformers can learn to discern protein contacts, despite never being trained on any geometric data. In addition to being practically applicable for contact prediction, this finding suggests that Transformer-like models that use attention are very well suited to the analysis of protein sequences – an insight which seems to have heavily inspired the architecture of AlphaFold2.

3.5 **AlphaFold: A Breakthrough**

AlphaFold, designed by Deepmind and entered in CASP14, is undoubtedly the most effective and successful protein structure prediction method of all time. On CASP14, AlphaFold’s predicted structures had a median RMSD of under 1 Å, and a GDT_TS of over .92, which is indicative of accuracy comparable with experimental structures. On the very hardest subset of CASP structures AlphaFold achieved a median of GDT_TS of .87 (indicating accuracy very close to experimental structures) [16]. For comparison, AlphaFold1 achieved a median GDT_TS of under .60 for similarly difficult targets in CASP13, and methods before AlphaFold1 achieved median scores of under .40. It is safe to say that, for a significant majority of unsolved protein structures, AlphaFold can provide predictions with sufficient quality for use in most downstream applications. Additionally, AlphaFold can reliably forecast the accuracy of its own predictions using various “confidence
metrics” that are outputted from the model, so biologists have a good idea of when AlphaFold’s structures can be trusted. AlphaFold is already being used to aid scientific discovery, and millions of AlphaFold’s predicted protein structures have been released in publicly-available databases. It is not an overstatement to say that AlphaFold is quite possibly the greatest achievement ever for machine learning and artificial intelligence.

How did AlphaFold achieve such remarkable progress? Interestingly, AlphaFold was not enabled any fundamental new insights about the biology or physics of the structure prediction problem. Indeed, all of the essential scientific insights (most importantly, the connection between coevolution and structure) were well-established before the design of AlphaFold1. It seems, then, that AlphaFold’s success was due to the use of a purpose-built neural network architecture with the correct inductive biases for reasoning about protein structures. In the remainder of this chapter we will describe the most important elements of AlphaFold’s architecture in more detail (omitting technical details but highlighting key ideas), and highlight important questions about protein structure prediction that remain unsolved despite this recent breakthrough.

At the heart of AlphaFold is a transformer-like neural network module dubbed the “Evoformer.” Unlike AlphFold1, which took as input a $N \times N$ grid of residue couplings from a precomputed statistical analysis of the MSA, AlphaFold takes as input the raw $M \times N$ MSA and analyzes it using the Evoformer [16].

Recall that, at each layer of a Convolutional Neural Network, an input of size $N \times N \times D$ is transformed into an output of size $N \times N \times D$ (assuming the input and output depths are constant). Similarly, a Transformer layer takes an input of size $N \times D$ and produces an output of size $N \times D$. In both of these models, the size $N$ of the tensors being transformed depends on the input data.

The layers of the AlphaFold Evoformer also transform blocks of variably-sized data, except the data representations are specially engineered for the analysis of proteins. Each layer of the AlphaFold Evoformer takes in two inputs: one $M \times N \times D$ “MSA representation,” and a $N \times N \times D$ “pair representation.” It then transforms these two representations to create outputs of the same dimensions, which are propagated to the next Evoformer layer.

AlphaFold takes in a number of different inputs, which it uses to populate the initial MSA and pair representations at the first layer of the Evoformer. These inputs are the
length-$N$ target amino acid sequence, the $M \times N$ raw MSA, and up to four template protein structures. The templates are experimentally solved protein structures with sequences that are similar to the target, which AlphaFold can use to inform its structure predictions.

At the first layer of the Evoformer, the MSA representation is essentially initialized with the raw $M \times N$ MSA, where each column represents a sequence position and each row represents a homolog from a different organism. The pair representation is initialized with a sum of several 2D inputs. First, each $(i, j)$ entry is initialized with a vector embedding of the $i$th target amino acid and the $j$th target amino acid. Information from the template structures is also injected into the pair representation. In particular, the Cβ distogram for each template and the tiled template amino acid sequences are embedded and added to the pair representation. In addition, the backbone and side chain torsion angles for each template are embedded and concatenated as additional rows in the initial MSA representation.

After the initial MSA and pair representations have been populated, these representations are transformed by successive Evoformer layers. Each Evoformer layer performs a series of attention-based updates to each representation. In essence, each row of the MSA representation preforms a attention update like in the Transformer, except the matrix of attention weights is biased by weights derived from the pair representation. Subsequently, each column of the MSA representation performs its own attention update to produce the MSA representation for the subsequent layer. To update the pair representation, a special attention update is used to propagate information between “triangles” of related positions of the form $(i, j), (j, k), (i, k)$.

After the final MSA and pair representations have been computed, they are used to predict the atomic coordinates of the protein structure, as well as a number of auxiliary outputs. The final atomic positions are computed using a computational scheme referred to as Invariant Point Attention (IPA). In IPA, the backbone of the protein is represented as a set of rigid triangles formed by the bonded $N - C\alpha - C$ atoms of each residue. Notably, these triangles are modelled as floating in an unconnected “gas.” This means that triangles connected by peptide bonds are not enforced to be close together – these constraints are learned by the model.

In more detail, the input to an IPA layer is a cloud of rigid triangles floating in space, each with a 3D position and orientation, along with a vector representation attached to
each triangle. The vector representations for each triangle perform a transformer-like attention update, except the pairwise attention weights are also biased by a learned term that incorporates the relative positions and orientations of the two triangles. In addition, the attention weights are biased by the final pair representation. After the representations are updated at each IPA layer, the vector representation at each triangle is used to compute a position and orientation update (represented in the local reference frame defined by that triangle), as well as a series of side chain torsion angles. This process is repeated until a final structure prediction is produced. At the beginning of the structure module, all of the triangles are initialized in the same orientation at the origin, and the representation vectors are initialized as the vectors from the first row of the final MSA representation. Note that, because the computation only uses relative distances and orientations, it is equivariant, meaning that a rigid translation or rotation of the input triangles to an IPA layer will result in the same transformation being applied to the output. Such symmetries are often desirable when designing learning algorithms for 3D-structured data.

Some of the auxiliary outputs produced by AlphaFold include a predicted $\alpha$-LDDT score for each residue, as well as a predicted TM Score for the entire model. These outputs are computed as linear projections of the final pair representation and the final structure module representations. Empirically, these confidence metrics are well-correlated with the corresponding ground-truth accuracy metrics computed on the output structure. AlphaFold also outputs a predicted $\beta$ distogram, which is used to create an additional loss for model training.

Finally, AlphaFold’s own outputs can be fed back into the model for further refinement in a process called “recycling.” This works by taking the first row of the final MSA representation and using it to initialize the first row of the MSA representation in the next iteration, as well as taking the output $\beta$ distance matrix and the final pair representation and using them to initialize the pair representation of the next iteration.

### 3.6 Remaining Challenges

As stated before, AlphaFold is an extremely impressive breakthrough in protein structure prediction, and it may rightly called a solution to the problem in many cases. However, some additional questions and challenges remain. First, like all coevolution-based meth-
ods, AlphaFold remains dependent on having a reasonably large and diverse MSA. As will be elaborated upon in the next chapter, AlphaFold produces far less accurate predictions when supplied with a single sequence as opposed to an entire MSA (i.e., the input MSA representation has dimensions $1 \times N$). Underscoring this point, the AlphaFold authors examined AlphaFold’s performance on new protein structures with geometries that had never been seen before in previously solved structures, and found that AlphaFold only failed when the MSAs were small (containing 20 sequences or fewer) [16].

Efforts to develop protein structure predictors with less dependence on MSAs are already underway. One model called RGN2 directly predicts the backbone $\phi$ and $\psi$ angles from a single protein sequence, and uses vector embeddings from a Masked Language Modeling transformer as an additional source of information [5]. RGN2 is competitive with AF2 in cases where MSAs are not available, but it still significantly underperforms AlphaFold in most cases.

Thus, it is still an open question whether we can predict protein structures from single amino acid sequences based on the physical principles of folding alone (in a way that scales to large proteins and uses reasonable computational resources). Such methods will allow for the prediction of protein structures even when substantive MSA information is not available (i.e., for proteins with few known homologs in other species). Developing accurate and efficient structure prediction methods that are based on physical principles may also have other benefits – for instance, predicting the structural effects of single amino acid substitutions. These open questions are addressed by the original contributions of this thesis, which we will be covering in the next chapter.
4.1 INTERPRETABILITY IN ALPHA FOLD: QUESTIONS AND HYPOTHESES

Given AlphaFold’s success and the theoretical possibility of predicting protein structures from single sequences, it is natural to ask why AlphaFold is dependent on MSAs for its accuracy. As mentioned in the introduction, we hypothesized that AlphaFold has learned an accurate biophysical potential function for assessing the accuracy of candidate protein structures, but needs the coevolutionary information in the MSA to locate an approximate global minimum in this potential function and solve the difficult search problem posed by Levinthal’s paradox. After finding the neighborhood of the global minimum using the MSA, the later stages of the AlphaFold model may descend the learned potential to produce a refined structure prediction. As described in the previous chapter, AlphaFold outputs a variety of confidence scores related to the predicted accuracy of its structures, and these confidence scores may correspond to the value of its internal potential function. This hypothetical prediction mechanism is illustrated in Figure 4.1.

In this section, we provide strong evidence that this hypothesis is a correct description of AlphaFold’s prediction workflow. Furthermore, we show that AlphaFold’s learned potential function is much more accurate, smoother, and more computationally efficient to compute than Rosetta and other previously developed potential functions, despite not using
any coevolutionary information. This finding has important implications for protein structure prediction without the need for MSAs. While AlphaFold is reliant on MSAs to solve the search problem of finding an optimum in its learned potential function, the AlphaFold model only has limited time and space in which to perform this search. Thus, by intelligently optimizing input structures within AlphaFold’s potential function, it may be possible to accurately predict protein structures without the need for MSAs, at the cost of investing more computation time than the default AlphaFold model.

4.2 Ranking Decoy Structures with AlphaFold

The hypothesis that AlphaFold has learned a physical potential function that does not depend on coevolution information is theoretically appealing, and lends itself to experimental testing. There are several avenues through which candidate protein structures can be introduced to the AlphaFold model. First, candidate structures can be supplied as templates, which AlphaFold uses to incorporate known structural information from proteins that are related to the target sequence. Second, candidate structure information can be introduced through AlphaFold’s recycling mechanism, which is normally used to supply previous model predictions back into the model for further refinement. Ideally, when a candidate structure is introduced through either of these mechanisms, we would hope that AlphaFold’s confidence metrics are closely correlated with the actual accuracy of the candidate structure, even when no coevolutionary information is supplied. If this is the case,
it suggests that AlphaFold has learned an accurate potential function of protein structures which does not rely on coevolutionary information. In our experiments, we show that this is indeed the case for candidate protein structures introduced as templates, and that the potential function given by AlphaFold’s confidence metrics outperforms previous state-of-the-art models at ranking protein structures, even when no coevolutionary information is provided.

4.2.1 METHODS

Computational biologists have historically predicted protein structures based on related sequences with experimentally solved structures (recall our earlier example of homology modelling in the Rosetta prediction workflow)[4]. As mentioned in the previous chapter, AlphaFold incorporates this approach into its workflow by allowing the structures of up to four related proteins to be supplied to the model as templates. For each template, AlphaFold receives the template’s one-hot-encoded amino acid sequence, Cβ distance matrix, and backbone and side chain torsion angles as inputs. In addition, AlphaFold is given a mask indicating which atoms are unresolved in the template structure, and ignores torsion angles involving those atoms. Recent papers have demonstrated that AlphaFold’s template mechanism can be used to refine structural hypotheses derived from experimental data or protein complex modelling [11, 41].

We investigated whether AlphaFold has learned a coevolution-independent potential function for scoring protein structures by supplying AlphaFold with a.) a target amino acid sequence to be predicted and b.) a “decoy structure” that is passed to the model as a template. The goal of this procedure is to score the plausibility of the target amino acid sequence adopting the geometry given by the decoy structure. It is motivated by the hypothesis that AlphaFold’s output structure will closely resemble the decoy introduced as a template and therefore, if AlphaFold has learned an accurate potential function that does not require coevolution information, the output confidence metrics will closely track the quality of the decoy. Note that no coevolutionary information is supplied to the model during this procedure.

We chose to mask out all non-backbone atoms (aside from Cβ) in the decoy structure, since we found this to improve the correlation between AlphaFold’s confidence metrics and
backbone-based accuracy scores, and forgoing side chain atoms as inputs means that decoy structures can be produced by generating backbone conformations without the relatively expensive step of determining side chain placements. Results from decoy ranking with side chains are presented later in this chapter.

The decoy’s one-hot-encoded amino acid sequence may have an important effect on how it is processed by AlphaFold. For instance, the presence of a template with high sequence identity to the target sequence may result in the model copying the template structure with high confidence, even if the template structure is a relatively low-quality decoy. On the other hand, it is possible that AlphaFold will ignore any template structure that does not have high identity to the target sequence. We investigated two choices for the one-hot-encoded amino acid sequence associated with the decoy: the target amino acid sequence, and a sequence of all alanines. Note that, because we masked out all non-Cβ side chain atoms, both of these choices are consistent with the structural data being supplied to AlphaFold.

After processing its inputs, AlphaFold produces an output structure and a series of confidence metrics, including the predicted TM Score (pTM) and the predicted LDDT-Cα Score (pLDDT) [23, 45]. To determine whether AlphaFold has learned a MSA-free potential function for assessing protein structure accuracy, we investigated whether we could accurately rank the decoy structures based on AlphaFold’s outputs. For each decoy, we computed a “composite confidence score” by multiplying the output pLDDT, the output pTM, and the TM Score between the decoy structure and the AlphaFold output structure. The last term adjusts for the fact that AlphaFold’s confidence metrics ultimately reflect the accuracy of the output structure (which can differ from the decoy structure), while we were interested in scoring the decoy structures for the sake of direct comparison with other decoy-ranking methods. We found that AlphaFold’s output structures usually improved upon the quality of the decoy structures (as is illustrated in Figure 4.9), which supports the idea that AlphaFold can perform local structure refinement. Across our evaluations, we found that a composite score using both the pLDDT and pTM showed higher correlation with decoy quality than either metric in isolation.

We also attempted to inject candidate structure information through the recycling mechanism, but this technique did not produce consistent results. This may be because the recycling mechanism is designed to jointly take structural information and internal represen-
ations from previous iterations into the model, while we only supplied structural information.

4.2.2 Results: Rosetta Decoy Dataset

Using the procedure outlined above, we aimed to determine whether AlphaFold’s outputs could be used to assess the accuracy of decoy structures introduced as templates. For our initial evaluation we used the Rosetta decoy dataset, which contains 133 native protein structures with thousands of decoys corresponding to each native structure [32]. We compared AlphaFold’s ability to assess the quality of decoy structures with the Rosetta energy function, as well as DeepAccNet, which is a state-of-the-art machine learning model for estimating the accuracy of protein structure models [12]. All reported results are from AlphaFold model 1 with 1 recycle (i.e., a total of two iterations through the model).

We found the correlation between the composite confidence score and decoy quality to be robust and consistent, regardless of the decoy’s amino acid sequence. The average Spearman rank correlation between the composite confidence score and the quality of the decoy (as measured by TM Score to the native structure) was .923 when using an all-alanine sequence and .908 when using the target sequence, compared to average correlations of .831 and .760 for DeepAccNet and the Rosetta energy function. The AlphaFold-based metrics showed higher correlations with decoy quality than DeepAccNet and Rosetta on almost every target in the dataset. More details regarding rank-order correlations are presented in Figure 4.2. This greatly increased correlation with decoy quality suggests that the potential surface learned by AlphaFold is both smoother more accurate than the potential functions generated by Rosetta and DeepAccNet.

In addition to assessing rank-order correlations, another practical indicator of decoy-ranking performance is the quality of the top-ranked decoy for each target. This metric corresponds to the accuracy of a structure prediction workflow in which a number of candidate structures are generated and scored, with the highest-scoring structure becoming the final prediction. On the Rosetta decoy dataset, the top-ranked decoys selected via the composite AlphaFold confidence score had an average TM Score of .924 for the all-alanine sequence and .931 for the target sequence, compared to .917 for DeepAccNet and .901 for Rosetta. More details on top-1 accuracies are given in Figure 4.3.
Figure 4.2: AlphaFold confidence metrics show robust correlations with decoy quality. A.) Spearman correlation between AlphaFold composite score (using an all-alanine decoy sequence) and decoy TM Score vs. Rosetta Spearman correlation between Rosetta energy and decoy TM Score. Each point is a target in the Rosetta decoy set. B.) Same as (A), except comparing against DeepAccNet. C.) Same as (A), except with the decoy sequence set to the target sequence. D.) Same as (B), except with the decoy sequence set to the target sequence. E.) Median Spearman correlations between various metrics and decoy TM Score. Error bars are bootstrap 95% confidence intervals of the median. F.) Same as (E), except using the mean.
Figure 4.3: AlphaFold’s top-ranked structures have higher quality than top-ranked structures from other models. A.) Comparison of TM Scores for the decoy with the highest AlphaFold composite score (using an all-alanine decoy sequence) vs. the decoy with the best Rosetta energy. B.) Same as (A), except comparing against DeepAccNet. C.) Same as (A), except with the decoy sequence set to the target sequence. D.) Same as (B), except with the decoy sequence set to the target sequence. E.) Median TM Scores of the top-ranked decoys for various ranking metrics, as well as the median TM Score of AlphaFold’s prediction with no MSA. Error bars are bootstrap 95% confidence intervals of the median. F.) Same as (E), except using the mean.
Overall, these evaluations indicate that AlphaFold can assess the quality of candidate protein structure with state-of-the-art accuracy, even when no coevolution information is provided. It should be noted that AlphaFold’s structure predictions were of low quality when no templates were provided (average TM Score of .402), as is illustrated by the examples in Figure 4.4. Yet despite being unable to predict the structures of these proteins without an MSA, AlphaFold achieved excellent performance assessing the quality of decoys without any MSA inputs. This provides evidence for the hypothesis that AlphaFold has learned a potential function that is largely independent of coevolution information, but needs coevolution information to search for global optima in this potential.

4.2.3 Effect of the Decoy Sequence

As mentioned previously, we investigated two choices for the decoy’s one-hot-encoded amino acid sequence: a sequence of all alanines, and the target sequence. Both of these choices yielded highly accurate results for decoy ranking on the Rosetta decoy dataset, but there were some noteworthy differences between the two sequence choices. The all-alanine sequence achieved better rank-order correlations, while using the target sequence achieved better top-1 accuracies. Examining individual examples suggested that this was because the all-alanine sequence achieved better performance on ranking low-quality decoys, while the target sequence was better able to rank high-quality decoys. We hypothesize that, when using the target sequence, AlphaFold’s confidence metrics may be more correlated with the physical realism of local features of the fold, since the global geometry of the template is assumed to be relatively accurate due to the high identity between the template and the target sequence. In contrast, the all-alanine sequence has low identity to the target sequence, which may cause AlphaFold’s confidence metrics to depend more strongly on the global features of the fold. A detailed comparison of the two sequence choices, as well as an illustrative example, are given in Figure 4.5.

We investigated whether it was possible to combine the low-end accuracy of the all-alanine sequence and the high-end accuracy of the target sequence into a hybrid ranking method. To this end, we fit a simple logistic model to compute a weighted sum between the all-alanine confidence score and the target sequence confidence score. More specifically, the hybrid confidence score is computed as follows:
Figure 4.4: Model confidence corresponds closely to decoy quality. A.) Plots of AlphaFold composite confidence score (using the hybrid method) vs. decoy TM Score across various decoy template inputs for three example proteins. B.) Visualizations of the example proteins from (A) with various template inputs: no template, a medium-quality decoy, and the native structure. Color indicates model confidence.

\[
\alpha = \frac{1}{1 + \exp\left(-\frac{(S_A - 50)}{8}\right)}, \quad S_H = (1 - \alpha)S_A + \alpha S_T
\]

Where \(S_A\), \(S_T\), and \(S_H\) are the composite scores using the all-alanine, target sequence, and hybrid methods, respectively. Intuitively, this function is designed to give more weight
Figure 4.5: Comparison between all-alanine decoy sequence and target sequence. A.) Comparison of Spearman correlation between AlphaFold composite confidence score and decoy TM Score when using an all-alanine decoy sequence vs. Spearman when using the target sequence. B.) Comparison of top-ranked decoy TM Score when using an all-alanine decoy sequence vs. top-1 TM Score when using the target sequence. C.) AlphaFold composite confidence score (using an all-alanine decoy sequence) vs. decoy TM Score for PDB 1T2P. D.) Same as (C), except using the target sequence.

to the prediction using the target sequence when the decoy has higher quality. The weights of the logistic function were tuned on the Rosetta decoy set to maximize both Spearman correlations and top-1 accuracies, as well as to eliminate instances where our AlphaFold-based rankings were significantly outperformed by Rosetta or DeepAccNet. Decoy ranking results from the hybrid confidence score are presented in Figure 4.6. We found that this simple hybrid approach was able to outperform both the all-alanine sequence and target sequence.
Figure 4.6: A hybrid approach achieves better decoy-ranking performance than using an all-alanine sequence or the target sequence. A.) Spearman correlation between AlphaFold composite score (using the hybrid method) and decoy TM Score vs. Spearman correlation between Rosetta energy and decoy TM Score. B.) Same as (A), except comparing against DeepAccNet. C.) Comparison of TM Scores for the decoy with the highest AlphaFold composite score (using the hybrid method) vs. the decoy with the best Rosetta energy. D.) Same as (C), except comparing against DeepAccNet. E.) Mean Spearman correlations between various metrics and decoy TM Score. Error bars are bootstrap 95% confidence intervals of the mean. F.) Mean TM Scores of the top-ranked decoys for various ranking metrics, as well as the mean TM Score of AlphaFold's prediction with no MSA. Error bars are bootstrap 95% confidence intervals of the mean.
4.2.4 Results: CASP14 Evaluation

Although our results on the Rosetta decoy dataset are promising, there is a risk that the comparison between methods may be unfair, since these proteins may have been present in the training data of some of the models we have evaluated. To assess the decoy-ranking ability of AlphaFold on a novel sample of proteins, we performed an additional evaluation on the Estimation of Model Accuracy (EMA) task from CASP14 [19]. For consistency with reported accuracy metrics from CASP, we measured protein structure quality using GDT_TS instead of TM Score for this evaluation [44].

To set up the CASP14 EMA experiment, the CASP organizers created a set of decoy structures by taking the 150 most accurate automated server submissions for each structure prediction target in CASP14. It should be noted that the decoy set does not include predictions from AlphaFold, since AlphaFold was entered in CASP14 as a “human group” rather than a server.

We replicated this evaluation using AlphaFold to assess the decoy structures, and compared the results with DeepAccNet (entered in in CASP as BAKER-experimental) and DeepAccNet-MSA (entered as BAKER-ROSETTASERVER), which were two of the best-performing EMA methods from CASP14 [19]. DeepAccNet-MSA is similar in architecture to DeepAccNet, except it also uses coevolution data for structure accuracy assessment.

The CASP assessors evaluated EMA methods based on their top-1 GDT_TS loss, which is defined as the difference in GDT_TS scores between the best decoy and the top-ranked decoy by a given EMA method. EMA methods were ranked based on their average GDT_TS loss over targets where at least one decoy had GDT_TS over 0.4, as well as the average Z-Score of their GDT_TS loss over these targets. For both of these metrics, the AlphaFold composite confidence score significantly outperforms all other EMA methods entered in CASP14. When comparing directly to DeepAccNet and DeepAccNet-MSA, the AlphaFold-based ranking performs better on a majority of targets. A detailed comparison of models on the CASP14 evaluation is presented in Figure 4.7. Note that, in the CASP14 EMA experiment, methods were also evaluated on their ability to rank the all-atom LDDT score of the decoy structures. We did not perform this comparison, since AlphaFold’s confidence metrics only reflect Ca-based accuracy metrics, and we omitted side chain information from our decoy proteins.
Figure 4.7: AlphaFold outperforms the top models from the CASP14 EMA experiment, even with no co-evolution information. A.) Comparison of GDT_TS scores for the decoy with the highest AlphaFold hybrid composite score vs. the decoy with the highest DeepAccNet score. B.) Same as (A), except comparing against DeepAccNet-MSA. C.) Spearman correlation between AlphaFold hybrid composite score and decoy GDT_TS vs. correlation between DeepAccNet score and decoy GDT_TS. D.) Same as (C), except comparing against DeepAccNet-MSA. E.) Average GDT_TS loss for top EMA methods entered in CASP14. For AlphaFold we report performance using the hybrid method, the all-alanine sequence, and the target sequence. For the all-alanine sequence and the target sequence, we show the performance when ranking decoys using the composite score with just pLDDT, just pTM, and both. Error bars are bootstrap 95% confidence intervals of the mean. F.) Same as (E), except examining the average Z-Score of GDT_TS loss for top EMA methods entered in CASP14. Note that these rankings are slightly different from the official ones published by CASP because we only assessed GDT_TS-based ranking, instead of GDT_TS and LDDT.
These results indicate that AlphaFold can reliably assess the accuracy of candidate protein structures without the use of coevolution information. However, coevolution data (or a method that can generate decoys close to the correct structure) are still necessary for accurate structure prediction. When AlphaFold is tasked with predicting the CASP14 targets without any MSA inputs, its structure predictions are generally much less accurate than the top-ranked decoy based on AlphaFold’s confidence metrics (Figure 4.8).

Figure 4.8: Without coevolution information AlphaFold generally fails to produce accurate predictions on CASP14 targets, but still achieves state-of-the-art performance at ranking decoys.

4.2.5 Relationship Between Input and Output Structures

As mentioned previously, AlphaFold’s output structures can differ from the structures provided as templates. Figure 4.9 illustrates that AlphaFold’s output structures are often similar in quality to the decoy structures, and sometimes are substantially improved in terms of TM Score and GDT_TS. This necessitates the need for a term in the AlphaFold composite score that tracks how much the AlphaFold output structure changes from the decoy structure, since AlphaFold’s confidence metrics ultimately reflect the accuracy of the out-
Figure 4.9: Comparison between input and output structure qualities. A.) TM Score of AlphaFold output structure vs. TM Score of decoy structure supplied as template with an all-alanine sequence. Each dot is single decoy in the Rosetta decoy set, color indicates composite confidence score. B.) GDT_TS of AlphaFold output structure vs. GDT_TS of decoy structure supplied as template with an all-alanine sequence. Each dot is single decoy in the CASP14 EMA set, color indicates composite confidence score. C.) Same as (A), but using the target sequence for the decoy. D.) Same as (B), but using the target sequence for the decoy. E.) Mean TM Scores of the top-ranked Rosetta decoys for various ranking metrics, including the AlphaFold output structures with the highest pLDDT × pTM product in blue. Error bars are bootstrap 95% confidence intervals of the mean. F.) Same as (E), except using the CASP14 EMA set. Note that this plot shows raw top-1 GDT_TS rather than GDT_TS loss.
put structure. As illustrated in Figure 4.9, applying this correction causes the confidence score to track the quality of the input (i.e., the color gradient in Figure 4.9 progresses along the x-axis) rather than the quality of the output (the y axis). It is interesting to note that, while AlphaFold is in many cases capable of improving decoy structures without coevolution information, it generally fails to predict these structures from scratch when no coevolution information is provided. This supports the idea that AlphaFold can perform local optimization over its learned protein potential, but needs coevolution data or a template to locate an approximate starting point for this optimization.

4.2.6 Results with Side Chains

In our experiments, we primarily considered ranking decoys based on their backbone geometry alone. We did this by masking out all non-backbone atoms aside from Cβ in the decoy structures that were provided to AlphaFold as templates. This makes sense because our goal was to rank decoys based on the accuracy of their backbones, and AlphaFold’s confidence outputs were trained to predict Cα-based accuracy metrics. In addition, it is relatively straightforward to generate candidate backbone geometries, so ranking structures from backbones alone makes it easier to search for high-confidence decoy structures. We chose to include Cβ atoms because their positions are fully determined by the backbone, and AlphaFold embeds the geometry of the template using a Cβ distance matrix.

In Figure 4.10, we show decoy ranking results with side chain geometries included in the decoys. When including side chain information, the decoy’s one-hot-encoded amino acid sequence was set to the target sequence in order to ensure consistency with the structural information provided to AlphaFold. We found that, across all of our evaluations, including side chain information resulted in somewhat lower correlations between AlphaFold’s confidence metrics and decoy quality, especially for lower-quality decoys. This is consistent with our previous observation that setting the decoy sequence to the target sequence resulted in lower correlations that using an all-alanine sequence. It is also consistent with our hypothesis that adding more detailed features to the template (like an informative sequence or side chain geometries) causes AlphaFold to pay more attention to local features. Because of the lower correlation for low-quality decoys, including side chain geometries results in lower performance on the CASP14 EMA experiment, although AlphaFold still
outperforms all other entrants when pLDDT and pTM are combined.

**Figure 4.10:** Results from ranking decoys with side chains included. A.) Comparison of Spearman correlation between AlphaFold composite confidence score and decoy TM Score when using an all-alanine decoy sequence vs. Spearman when using including the side chains. B.) Comparison of Spearman correlation between AlphaFold composite confidence score and decoy TM Score when using the target sequence vs. Spearman when including the side chains. C.) Average GDT_TS loss for top-performing methods in the CASP14 EMA experiment, including AlphaFold’s decoy rankings when side chain information is included in the decoy structures. D.) Same as (C), except using the average Z-Score of the GDT_TS loss.
4.3 Discussion

In this section we have demonstrated that AlphaFold has learned a protein structure potential which does not need coevolution information to achieve high accuracy, although AlphaFold still needs coevolution data to search for global minima in this potential. While this potential function achieves state-of-the-art performance on ranking decoys, it is not perfect, and is outperformed by other ranking methods on some targets. Still, this finding has significance for the interpretation of protein structure prediction models, as well as practical applications.

One such application is the prediction of protein structures when MSAs are not available. We have found that, even without coevolution information, AlphaFold’s confidence metrics closely track the quality of decoy structures injected as templates. This suggests a method for MSA-free protein structure prediction: search over the space of decoy structures to optimize AlphaFold’s output confidence metrics. While the space of potential decoy structures is too vast to search exhaustively, searching the latent space of a generative model with knowledge of plausible protein folds could make this task more feasible. Our decoy-scoring approach has the potential to make this process especially efficient, since AlphaFold achieves state-of-the-art decoy ranking performance without requiring side chain structural information. This reduces the search space of potential structures, and allows for the use of efficient backbone conformation generators. For example, the Rosetta de novo protein structure prediction method generates decoys by stitching together backbone torsion angles from fragments of solved proteins, and then uses simulated annealing to search over the space of decoys and optimize an empirically-derived scoring function [37]. Replacing this scoring function with one based on AlphaFold would likely make such methods far more accurate. Alternatively, fast machine-learning-based backbone generators (like the one used in RGN2) could be used to create candidate backbones, and then those backbones could be optimized for AlphaFold confidence scores using reinforcement learning (see the following chapter for initial exploration in this direction). Future research should extensively explore these approaches, and compare their effectiveness with other protein structure prediction approaches like RGN2 that do not directly utilize MSAs. In summary, our experiments show that AlphaFold has learned a highly accurate physical potential of protein structures that does not require coevolution information, which has exciting impli-
cations for the future of physics-based protein structure modelling. Some applications of this potential function to important problems in protein science are explored in the next chapter.
In the previous chapter, we demonstrated that AlphaFold has learned a highly accurate biophysical potential function for assessing the accuracy of protein structures. This result is exciting, since it indicates that AlphaFold has learned some principles of the physics of protein folding, and opens the door to a variety of new applications. This chapter begins exploring some of those applications, and lays out directions for future research on these important problems.

5.1 MSA-Free Structure Prediction

As mentioned previously, one potential application of AlphaFold’s learned potential function is the prediction of protein structures from single sequences, without the need for MSAs. As indicated by AlphaFold’s state-of-the-art top-1 accuracy metrics on the Rosetta dataset and CASP, finding a decoy structure which gives a very high AlphaFold confidence score is a reliable indicator of that decoy’s accuracy. Furthermore, AlphaFold’s confidence metrics are far more correlated with decoy qualities than confidence metrics from other models, indicating that the AlphaFold’s learned potential landscape may be relatively
smooth, raising the prospects that iterative optimization will converge on high-confidence decoys. Finally, AlphaFold’s potential function achieves state of the art accuracy while only taking backbone conformations as inputs. In contrast, other backbone-only potential functions, such as the coarse-grained Rosetta function, have shown far lower accuracy than their all-atom counterparts. This means that computationally efficient backbone generators can be used to rapidly generate decoys, making the search process more efficient.

There are many possible approaches to searching for high confidence decoys. As alluded to earlier, it should be possible to adapt the MCMC protocol used by Rosetta’s structure prediction workflow by replacing the coarse-grained Rosetta energy function with an AlphaFold-based confidence score. Another approach is to search for high-confidence decoys in the latent space of a generative model that outputs protein structures. The advantage of this method is that it should constrain the search space to plausible biological protein structures. We have explored a simple version of this approach, and find that it shows promising performance.

As opposed to building a generative model for protein structures from scratch, we decided to generate candidate structures using AlphaFold itself. Recent work has applied protein structure prediction models to fixed-backbone protein design (i.e., the design of a protein sequence that folds into a specific shape) by backpropogating a loss function of a predicted structure back to the model’s input sequence, and descending the gradient to modify the input sequence [31]. This suggests an approach to sampling candidate protein structures that is reminiscent of a generator-discriminator neural network architecture, where the generator and discriminator are two distinct instances of the AlphaFold model. The generator takes a randomly-initialized amino acid sequence as input, and produces a protein structure as output. The coordinates of this structure are then turned into a template and injected into the discriminator model, which is tasked with predicting the structure of the target amino acid sequence. The discriminator will then output a prediction and a confidence score. Our previous experiments illustrate that the confidence score is highly correlated with the quality of the injected template structure. To take advantage of this correlation, we backpropagate the confidence signal all the way back to the input sequence and update it via gradient descent. The hope is that this causes the input sequence to be molded such that it causes the generator to produce a high-quality template, and therefore a high confidence score from the discriminator. Ideally, the high-quality template
output from the generator and the subsequent output from the discriminator would have
greater MSA-free accuracy than the standard AlphaFold model. We implemented this ap-
proach and tested it on proteins from the Rosetta decoy data set.

Figure 5.1: Attempts at optimizing a candidate structure for PDB 1MJC. A.) Left: outputs before optimiza-
tion. Middle: change in confidence metrics over the course of optimization. Right: outputs after optimization.
This test used one recycle in the generator, no recycles in the discriminator, and no dropout. B.) Same as (A),
except using dropout in the generator. C.) Same as (A), except dropout in both models and random alterna-
tion between zero and one recycles in the generator.

While we were able to backpropograte gradients from the confidence scores to the input
sequence, we did not achieve high-quality results from this approach. One issue we encoun-
tered was the generation of adversarial structures, which give the discriminator high confidence when injected as templates, but which are not truly accurate structures. This adversarial behavior can be mitigated by adding more noise to the prediction process through measures like dropout in the generator and discriminator, as well as varying the number of recycles across optimization iterations. As more of these measures are applied, the adversarial behavior goes away, but optimization becomes increasingly unaffectionate. Examples of these effects are given in Figure 5.1.

In order to prevent the generation of adversarial decoy structures, we developed a new protocol in which we successively mutate the input sequence to the generator and select for sequences that give high confidence outputs from the discriminator, similar to a genetic algorithm. More specifically, we used the following procedure with the same generator-discriminator architecture as before. The input sequence to the generator model is initialized to the target sequence. At each round of optimization, 10 random mutations are introduced to the input sequence to produce 10 candidate sequences. Each candidate sequence is supplied to generator model to create a new candidate structure, which is then passed to the discriminator model as a template. The confidence outputs of the discriminator model are then used to rank the mutated candidate sequences, and the top-ranked candidate sequence is chosen to be the input sequence for the next round of optimization. After repeating this procedure for 20 iterations, we took the discriminator output with the highest confidence as our structure prediction. For computationally simplicity, we used no recycling iterations in the generator and discriminator.

We found that, on the Rosetta dataset, structure predictions from this workflow had significantly higher accuracies than AlphaFold’s default predictions with no MSA. Furthermore, this effect could not be explained by the fact that “stacking” the generator and discriminator has a similar effect to adding more recycles: The outputs from our genetic optimization procedure also have higher accuracies than AlphaFold’s default MSA-free prediction with three recycling iterations. These results are presented in Figure 5.2.
These exciting preliminary results confirm the intuition that AlphaFold’s learned physical potential can be used to search for high-accuracy candidate protein structures without MSAs. In our current sample of proteins, some targets fail to significantly improve, and remain in the realm of low-confidence and low-accuracy structures. Such cases could likely benefit from more sophisticated search strategies, which we will discuss at the end of this chapter. Still, our results on MSA-free structure prediction are an exciting step forward towards the goal of efficiently predicting protein structures from single sequences based on the physical principles of protein folding.

5.2 Mutation Effect Prediction

In addition to protein structure prediction, another important task in computational protein science is predicting the effects of mutations on protein structures and functions. Successful mutation effect prediction systems can be helpful for diagnosing whether genetic mutations will have serious health consequences, and for designing proteins with desired
functionalities. Recently, language models of protein sequences (like Masked Language Modelling Transformers) have achieved state-of-the-art performance at forecasting the effects of mutations [26]. As mentioned earlier, MLM Transformers are trained to predict the identity of the amino acid at a masked-out position in a protein sequence. Intuitively, a Transformer should assign higher probabilities to amino acids at the masked position that would give the protein higher evolutionary “fitness,” since fitter proteins are more likely to be observed in nature (and thus are more likely to be observed in the model’s training data). Therefore, the difference in the probabilities assigned to amino acids A and B at a given position gives an estimate of the fitness difference between a protein with amino acid A at that position, and a mutational variant with amino acid B at that position.

To assess the effectiveness of mutation effect prediction systems, it is typical to measure the correlation between predicted fitness changes and the results of so-called “deep mutation scanning” experiments. In such experiments, a large number of mutations are introduced into a protein, and the fitness of the resulting mutants in measured. One canonical example comes from the bacterial protein β-Lactamase, which is responsible for degrading the antibiotic ampicillin and thereby conferring bacteria with ampicillin resistance. To measure the effects of mutations on β-Lactamase, biologists have introduced a large number of mutations to the β-Lactamase gene, inserted the modified gene into bacteria, and then measured the growth of the mutant bacteria in an environment containing ampicillin. In this case, the “fitness difference” between two protein variants is the difference between the growth rates of bacteria containing those respective variants [39].

State-of-the-art fitness predictions based on Transformer models have achieved Spearman correlation coefficients of around 0.8 with experimentally-measured fitness in β-Lactamase mutants. The physical and thermodynamic stability of the native protein structure is undoubtedly a large contributor to the fitness of mutational variants. Therefore, the change in free energy associated with swapping the identity of an amino acid residue in the folded protein structure is likely a strong predictor of the fitness effect of that mutation. Biophysical energy functions like those computed by AlphaFold and Rosetta thus have the potential to be accurate mutation effect predictors.

To test whether this was the case, we ran AlphaFold with the target sequence set to the wildtype (i.e., unmutated) β-Lactamase sequence, and supplied the native structure of β-
Lactamase as a template. Then, we repeated this procedure with various mutations introduced to the target sequence, and recorded the change in AlphaFold’s confidence metrics relative to the wildtype target sequence. As we have demonstrated previously, AlphaFold’s confidence outputs in this setting are a strong indicator of the biophysical compatibility between the target sequence and the template structure. Therefore, if a particular mutation causes a large decrease in AlphaFold’s confidence, it may indicate that this mutation decreases the physical stability of the native structure.

After running this experiment, we found a moderate association between AlphaFold’s confidence metrics and the fitness of each mutational variant. In particular, we found a Spearman correlation of 0.55 between AlphaFold’s pTM output and the experimental fitness of each mutation. Results from this mutation effect prediction experiment are presented in Figure 5.3. This correlation is considerably lower than the correlation of .8 achieved by language models. However, it is not expected that a physics-based prediction of mutation effects should outperform such language models, since the language models can capture the effects of mutations that do not cause physical destabilization, but do cause decreases in fitness (for example, the mutation of key residues in the active site that are responsible for binding ampicillin). While these mutations do not cause a significant free energy change, their importance for fitness is signalled by their heavy conservation across evolutionary space. Language models trained across many evolutionarily related protein sequences can take this into account, while physics-based potentials are unaware of the biological significance of such mutations.

Nevertheless, AlphaFold still performs slightly worse than Rosetta on this dataset, with Rosetta’s predicted energy changes showing a correlation of 0.62 with experimental fitness measurements. This is interesting given that AlphaFold’s learned potential strongly outperforms Rosetta at ranking the accuracy of global protein structures. However, it is not altogether unsurprising. While the parameters of Rosetta were designed to capture protein physics from first principles, AlphaFold’s understanding of physics was learned in service of protein structure prediction. Therefore, while AlphaFold probably understands fundamental principles like the unfavorability of hydrophobic residues in the protein core, it may be less sensitive to violations of those principles by single amino acid residues, and more concerned with the general accuracy of the structure across the entire peptide chain.
Figure 5.3: Plot of experimental fitness measurements vs. AlphaFold pTM for β-Lactamase mutants. Color indicates RMSD-change of the predicted structure from the wildtype prediction, with lighter indicating lower change.

5.3 Discussion

In this chapter, we examined two applications for the biophysical potential learned by AlphaFold: the prediction of protein structures from single sequences without MSAs, and predicting the effects of mutational variants on proteins. In the first application, we showed that iteratively optimizing decoy structures to maximize AlphaFold’s confidence metrics can result in structure predictions that are significantly better than AlphaFold’s default predictions without MSAs. This result is especially impressive due to the simple search strategy used, which is limited to exploring the structures generated by AlphaFold’s predictions on the region of sequence space surrounding the target sequence. To expand upon this exciting preliminary result, more sophisticated search strategies should be considered. One more complex search strategy could involve reinforcement learning, in a setup that would reward an agent for generating decoy structures that give high confidence outputs from AlphaFold. An architecture like the backbone generator used in RGN2 could potentially be employed for this application. Reinforcement Learning has shown impressive results on optimizing rewards in open-ended search problems before (for example, in designing the layouts of circuit boards), so it may be particularly well-suited to this task [28].
We also investigated whether AlphaFold’s learned biophysical energy function can be used to predict the effects of mutations on protein structures and functions. We found a moderate association between AlphaFold’s confidence metrics and mutational fitness, although this association was somewhat weaker than the Rosetta energy function. Future work should expand this analysis to more proteins, and examine the specific types of mutations for which AlphaFold’s confidence outputs do and do not track mutational fitness. Such investigations may lead to better explanations of why AlphaFold’s physical potential produces mediocre results on mutation effect prediction, despite its success on global accuracy estimation.

Finally, one important application which we have not yet explored is the analysis of alternative protein conformations. It would be extremely interesting to sample decoys from a protein structure with multiple stable conformations, and investigate whether decoys from distinct conformations generate distinct modes in AlphaFold’s confidence landscape. If this is the case, and if those distinct conformations can be located through sampling schemes discussed previously, this may provide a principled method of predicting all of a protein sequence’s possible native structures, as opposed to the single native structure currently returned by AlphaFold.
6 Conclusions

In this thesis, we have explored the current state of research in protein structure prediction and presented new results addressing open questions in this field. We began by reviewing the biophysical principles that underlie protein folding and function, and subsequently reviewed important methods in protein structure prediction. These methods included physics-based approaches, which attempt to directly simulate the process of protein folding at various levels of abstraction, and coevolution methods, which take advantage of the correspondence between evolutionary variation and protein structure. Our discussion of protein structure prediction also emphasized how the combination of deep learning architectures with coevolution methods has led to rapid progress in recent years, culminating in the breakthrough performance of AlphaFold. Lastly, we highlighted open questions about protein structure prediction, including whether it is possible to predict protein structures from single sequences based on physical principles, and whether models like AlphaFold have any understanding of the physics of protein folding.

The original contributions of this thesis pursued these open questions through careful and thorough experiments involving AlphaFold. We demonstrated that AlphaFold achieves state-of-the-art performance at ranking the accuracy of candidate protein structures without the use of any coevolution information, even though it requires coevolution for accurate structure prediction. This finding supports the hypothesis that AlphaFold has learned
an accurate biophysical potential function for assessing the accuracy of protein structures, and uses coevolution information to find the approximate neighborhood of a global optimum in this potential function. After using the coevolution information to locate an approximate minimum in its learned potential function, it appears that the later stages of the AlphaFold model act as an “unrolled optimizer” that locally descends the potential function to produce a refined structure output. The fact that AlphaFold has learned a highly accurate potential function, which is relatively efficient to compute and does not require coevolution information or side chain geometries for its accuracy, opens the door to several exciting applications, including the accurate prediction of protein structures without using MSAs.

We explored this application by iteratively optimizing decoy structures to produce high confidence outputs from AlphaFold, and found that we were able to improve upon AlphaFold’s default MSA-free structure predictions. This exciting finding confirms that our work is a promising step on the road towards accurate protein structure prediction from single sequences, which is one of the main outstanding challenges in protein structure prediction. We also attempted to use AlphaFold’s potential function to predict the effects of mutational variants, and found that it achieved decent, but not outstanding, results. This outcome may be related to AlphaFold’s focus on predicting globally correct protein folds, rather emphasizing the correctness of individual residues.

Overall, our findings strongly support the notion that protein structure prediction models can learn physical principles underlying protein folding, and that their learned understanding of protein physics can be transferred to important downstream applications. This work opens up many interesting future research directions. Some of these directions, such as exploring more sophisticated sampling methods for MSA-free protein structure prediction, have already been described in previous chapters. Other more long-term directions could include the training of new AlphaFold-like models in a manner designed to improve their ability to learn accurate potential functions of protein structures. For example, it would be interesting to train an AlphaFold-like model in an adversarial setting, in which a discriminator model tries to distinguish between predicted structures and native structures, and a generator model tries to make structure predictions that fool the discriminator. The discriminator may then learn an even more accurate function for scoring model accuracy. In addition, native structures with side chain substitutions could be periodically supplied.
to the discriminator as decoys. Training the discriminator to identify such mutants as non-native structures may increase its ability to predict the functional and structural effects of point mutations.

Another important future direction concerns the ability of AlphaFold’s learned potential function to model protein dynamics. In the previous chapter, we hinted at the possibility of using AlphaFold’s confidence metrics to identify alternative conformations of proteins. Taking this idea a step further, it would be interesting to see if AlphaFold’s confidence measures correlate with the probability of seeing a particular protein conformation in an ensemble of poses generated via molecular dynamics simulation. If a strong correspondence exists, it would provide further evidence that AlphaFold has truly learned a physically accurate potential. If AlphaFold can be made to run fast enough, it may even be possible to run dynamical simulations of proteins with AlphaFold’s confidence outputs playing the role of a potential energy function, thus facilitating accurate coarse-grained simulations of protein motion.

An additional set of important applications lies in the design of new proteins. Designing proteins that fold into specific shapes and bind to specific targets is a quickly growing field of research, and holds enormous promise for the creation of novel therapeutics [14]. Such design problems are usually tackled using potential functions like Rosetta, but extra-accurate potentials like the one learned by AlphaFold may be able to improve this process. In the case of designing peptide binders, this will require extending AlphaFold’s potential function to encompass complexes between multiple proteins, which may be made possible by recently-developed AlphaFold-multimer models [9].

We will end on a note regarding the importance of understanding and appreciating the world of proteins. The struggles of the COVID-19 pandemic have brought protein science extremely close to home for billions of people around the world. The SARS-CoV-2 spike protein has become a symbol of our time. Functionally relevant amino acid substitutions in that protein have become world-famous, although they are often referred to by monikers like Alpha, Delta, Omicron, and BA.2. Our ability to model, manipulate, and synthesize the spike protein has been the basis for every COVID-19 vaccine in the world. Monoclonal antibody treatments against COVID-19 are themselves comprised of proteins, and other COVID-19 drugs are designed to target coronavirus proteins of interest. The efforts of protein researchers around the world have been crucial to our species’ response against the...
virus. The first structures of the spike protein were solved mere months after the virus was first identified, enabling rapid vaccine development [22]. As our understanding of proteins continues to progress, hopefully so too will our ability to use that knowledge to help people during the next pandemic and beyond.
Bibliography


This thesis was typeset using \LaTeX, originally developed by Leslie Lamport and based on Donald Knuth’s \TeX. The above illustration, *Science Experiment 02*, was created by Ben Schlitter and released under [CC BY-NC-ND 3.0](https://creativecommons.org/licenses/by-nc-nd/3.0/). A template that can be used to format a PhD dissertation with this look & feel has been released under the permissive AGPL license, and can be found online at [github.com/suchow/Dissertate](https://github.com/suchow/Dissertate) or from its lead author, Jordan Suchow, at suchow@post.harvard.edu.