EPFL

Thèse n° 10 271

Predicting protein interactions using geometric deep learning on protein surfaces

Présentée le 28 mars 2024

Faculté des sciences et techniques de l'ingénieur
Laboratoire de conception de protéines et d'immuno-ingénierie
Programme doctoral en biologie computationnelle et quantitative

pour l'obtention du grade de Docteur ès Sciences

par

Freyr SVERRISSON

Acceptée sur proposition du jury

Prof. M. Dal Peraro, président du jury
Prof. B. E. Ferreira De Sousa Correia, Prof. M. Bronstein, directeurs de thèse
Prof. P. Liò, rapporteur
Dr A. Loukas, rapporteur
Prof. Ph. Schwaller, rapporteur

École polytechnique fédérale de Lausanne

2024

Acknowledgements

The journey of this thesis has been both challenging and rewarding, and it would not have been possible without the unwavering support and guidance of many remarkable individuals.
Firstly, I owe a significant debt of gratitude to my advisor Prof. Bruno Correia. Your guidance has been invaluable throughout this journey. Your deep knowledge, combined with your encouraging approach, has been pivotal in shaping my research. You've consistently provided the right balance of direction and autonomy, allowing me to grow as an independent researcher. Your dedication to my success, both in terms of time and effort, has left a lasting impact on my academic journey. I'm truly grateful for all you've done.
I also want to acknowledge my co-advisor, Prof. Michael Bronstein. Your insights and expertise in our research domain have been beneficial. Your constructive feedback and questions often prompted me to think more critically about my work. Collaborating with you provided a different perspective that enriched the research process. I appreciate the guidance you've offered and the knowledge you've shared throughout this journey.

A special acknowledgment goes to the postdocs who played pivotal roles in shaping the direction and depth of my research, Pablo Gainza and Jean Feydy.

Pablo, it was under your initiative that the research direction of my thesis was born. Your vision and foundational work laid the groundwork for what would become the core of my studies. Your foresight in identifying and pursuing this avenue of research has been instrumental, and I'm grateful for the foundation you provided.

Jean, your technical prowess and deep knowledge were key in elevating our work. Whenever we faced challenges, your expertise often provided the clarity and solutions we needed. Your ability to delve deep into technical intricacies and refine our methodologies ensured our research was both rigorous and impactful.

Together, your combined insights and contributions have been central to the success and quality of my thesis. I'm genuinely grateful for the collaborative spirit and dedication both of you brought to our shared endeavors.

I'd like to acknowledge key collaborators from VantAI who were essential in the concluding phase of my research: Mehmet, Dylan, and Luca. Mehmet and Dylan, whose technical expertise significantly advanced our project, ensuring a blend of academic rigor with practical solutions. Luca, your management oversight and strategic direction streamlined our efforts,

ensuring alignment with broader objectives. The collaboration with each of you added depth and efficiency to our research, and I'm grateful for your roles in its success.
I must also express my gratitude to friends and colleagues who have been pillars of support and collaboration throughout my PhD journey. Special mention goes to Zander and Ahmet, who have been not just colleagues but true friends, always ready to lend a hand, share insights, or simply offer words of encouragement. Additionally, a nod to those who were with me at the very start of this journey: Andreas, Jaume, Sarah, and Sailan. Your guidance, early discussions, and shared experiences laid a strong foundation for what was to come. Each of you, in your unique way, has enriched my PhD experience, and I'm genuinely thankful for your presence and contributions.
Additionally, a word of appreciation for the newer members of our group. While our paths may have crossed later in my journey, your enthusiasm, fresh perspectives, and dedication have invigorated our collective efforts. A particular shoutout to the team in the Drylab. Your specialized skills and collaborative spirit have been instrumental in pushing the boundaries of our research. It's been a pleasure witnessing and being a part of the growth and dynamism you bring to the group.
A sincere thank you to Alice, who stood by me throughout the four years of my PhD. Your support and understanding during this journey have been invaluable.
Lastly, to my parents, your belief in me and your endless encouragement have been the foundation upon which all of this was built. Thank you for your love and the values you instilled in me, which have guided me throughout this journey.

Lausanne, 25 August 2023 F. S.

Abstract

In the domain of computational structural biology, predicting protein interactions based on molecular structure remains a pivotal challenge. This thesis delves into this challenge through a series of interconnected studies.

The first chapter introduces the concept of protein molecular surfaces, which are characterized by distinct patterns of chemical and geometric features, serving as fingerprints for their interaction modalities. We present MaSIF (Molecular Surface Interaction Fingerprinting), a novel geometric deep learning framework. This tool is adept at predicting protein pocket-ligand interactions, protein-protein interaction sites, and scanning protein surfaces for potential protein-protein complexes.

Building on the insights from the initial chapter, the second chapter addresses the limitations of mesh-based representations in protein structures. We propose a deep learning framework that computes and samples the molecular surface directly from the atomic point cloud. This method, which requires only raw 3D coordinates and chemical types of atoms as input, has demonstrated state-of-the-art performance in identifying interaction sites and predicting protein-protein interactions.

The third chapter, informed by the preceding work, presents DiffMaSIF, a cutting-edge score-based diffusion model tailored for rigid protein-protein docking. DiffMaSIF leverages a surface-based molecular representation, integrated into an equivariant network, to efficiently predict protein complexes. This approach surpasses contemporary ML methods and aligns with traditional docking tools, but with a significantly reduced number of generated decoys.

Collectively, the research in this thesis offers a series of methodologies that, while building on each other, individually contribute significant advancements to our understanding and prediction of protein interactions, paving the way for future work in protein function prediction and design.

Key words: Protein-protein interactions, protein structure, deep learning, geometric deep learning

Résumé

Dans le domaine de la biologie structurale computationnelle, la prédiction des interactions protéiques basée sur la structure moléculaire demeure un défi majeur. Cette thèse s'attaque à ce défi à travers une série d'études interconnectées.
Le premier chapitre introduit le concept des surfaces moléculaires des protéines, qui sont caractérisées par des motifs distincts de caractéristiques chimiques et géométriques, servant d'empreintes pour leurs modalités d'interaction. Nous présentons MaSIF (Molecular Surface Interaction Fingerprinting), un nouveau cadre de travail basé sur l'apprentissage profond géométrique. Cet outil est habile à prédire les interactions entre les poches protéiques et les ligands, les sites d'interaction protéine-protéine, et à scanner les surfaces des protéines pour les complexes protéine-protéine potentiels.
S'appuyant sur les connaissances du chapitre initial, le deuxième chapitre aborde les limites des représentations basées sur des maillages dans les structures protéiques. Nous proposons un cadre d'apprentissage profond qui calcule et échantillonne la surface moléculaire directement à partir du nuage de points atomiques. Cette méthode, qui ne nécessite que les coordonnées 3D brutes et les types chimiques des atomes comme entrée, a démontré une performance de pointe dans l'identification des sites d'interaction et la prédiction des interactions protéine-protéine.
Le troisième chapitre, éclairé par les travaux précédents, présente DiffMaSIF, un modèle de diffusion innovant basé sur le score, conçu pour le docking rigide protéine-protéine. DiffMaSIF exploite une représentation moléculaire basée sur la surface, intégrée dans un réseau équivariant, pour prédire efficacement les complexes protéiques. Cette approche surpasse les méthodes ML contemporaines et s'aligne avec les outils de docking traditionnels, mais avec un nombre considérablement réduit de leurres générés.
Collectivement, la recherche présentée dans cette thèse offre une série de méthodologies qui, tout en se basant les unes sur les autres, contribuent individuellement à des avancées significatives dans notre compréhension et prédiction des interactions protéiques, ouvrant la voie à des travaux futurs sur la prédiction et la conception de la fonction protéique.

Mots clefs : Interactions protéine-protéine, structure des protéines, apprentissage profond, apprentissage profond géométrique.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introduction

1.1.1 Background and Motivation

Proteins are essential macromolecules that carry out a wide array of functions critical to life. They participate in cellular processes such as metabolism, signaling, structure, and transport. The ability of proteins to perform these varied roles relies on their capacity to interact with other biomolecules including other proteins, nucleic acids, lipids, and small molecules. Elucidating these complex interaction networks is therefore fundamental to understanding biology in molecular detail. However, characterizing protein interactions through experimental techniques alone poses significant challenges. High-throughput experimental methods for detecting protein interactions, such as yeast two-hybrid screening,often suffer from high false positive and false negative rates. Furthermore, such techniques do not provide insights into the structural basis and biophysical forces governing interactions. On the other hand, detailed biophysical techniques like X-ray crystallography demand extensive time, resources, and sample quantities that limit throughput.

Computational methods hold great promise in accelerating research on protein interactions, complementing experimental approaches. By leveraging statistical models and biophysical simulations, computations can integrate diverse datasets, predict binding partners, estimate binding affinities, and model interaction dynamics. However, current computational techniques for predicting protein interactions also face limitations. Many rely heavily on evolutionary information, performing poorly on proteins lacking homology. Others like molecular docking are computationally demanding, struggling with protein flexibility. Additionally, machine learning techniques often yield "black-box" models lacking interpretability.

Recent advances in deep learning provide new opportunities to tackle the multifaceted challenge of modeling protein interactions. By learning from large-scale datasets and establishing intricate feature representations, deep neural networks could better capture the complexity of protein interactions. Geometric deep learning extends these techniques to non-Euclidean

protein structure data, capturing critical biochemical and conformational determinants of binding. Realizing the potential of deep learning for protein interactions demands tackling key challenges including noisy and scarce training data, model interpretability, and computational efficiency.

This thesis aims to push forward the frontiers of deep learning-based modeling of protein interactions. We develop novel geometric deep learning approaches that learn interpretable protein surface patterns to predict diverse interaction types. In constructing computationally efficient models that rely solely on three-dimensional structure, we provide tools to accelerate discovery even in the absence of evolutionary information. By releasing these models and datasets to the scientific community, we hope to catalyze future efforts at the intersection of computation and protein science.

1.1.2 The Interplay of Biology and Computation in Modern Research

Biology and computation, though traditionally viewed as distinct disciplines, have in recent years become increasingly intertwined. This interplay has ushered in an era of unprecedented discovery and innovation, especially in the realm of molecular biology.

In the past, the complexities of biological systems made it a laborious task to analyze and interpret data. Biological phenomena operate across a wide range of scales, from the atomic level to whole organisms, and involve intricate networks of interactions. Modeling such phenomena using traditional analytical or experimental methods can be cumbersome, if not impossible, due to the sheer complexity and high dimensionality of the biological data.

Enter computation, with its ability to handle large-scale data and perform complex calculations. Modern computational methods have transformed the field of biology, enabling researchers to model intricate biological processes, predict molecular interactions, and even simulate the evolution of entire ecosystems. This marriage of biology and computation has birthed the field of computational biology [122], a discipline that leverages computational techniques to solve biological problems.

A key area where the power of computation has been harnessed in biology is in the study of proteins. Proteins, being central to numerous biological functions, present a compelling area of study. However, the sheer complexity of protein structures and the myriad interactions they participate in makes them a challenging subject to analyze through experimental means alone.

By using computational models and algorithms, researchers can predict protein structures [112], study protein-ligand interactions[18], simulate enzymatic reactions [55], and much more. This computational approach not only complements traditional experimental methods but also opens up new avenues of research that would have been unimaginable a few decades ago.

A perfect illustration of this interplay between biology and computation is the use of machine learning and artificial intelligence (AI) in biological research. Machine learning, a subfield of AI, has seen rising application in computational biology. From predicting protein structures to identifying potential drug targets, machine learning is revolutionizing how we approach biological data [9].

However, despite these advances, there remain many challenges in harnessing the full potential of computation in biology. Computational models are only as good as the data they are trained on and the assumptions they are built upon. Noise in biological data, scarcity of well-annotated datasets, and the dynamic nature of biological systems pose significant challenges. Furthermore, translating computational predictions into tangible biological insights is not always straightforward and requires deep biological understanding.

This thesis sits at the nexus of biology and computation, leveraging advanced computational methods, specifically deep learning, to tackle fundamental questions in protein biology. Our work serves to underscore the value of interdisciplinary approaches in modern research, illuminating how computation can be employed to elucidate biological complexity, particularly in the realm of protein interactions.

1.2 Basic Principles of Molecular Biology

1.2.1 Proteins: The Building Blocks of Life

Proteins, fundamental components of life, perform an array of functions central to every biological process [98]. These complex biomolecules, composed of one or more amino acid chains, serve varied roles within and outside cells, from providing structural support to catalyzing chemical reactions.

Protein structures, assembled in a hierarchical fashion, consist of primary (sequence of amino acids), secondary (local structures like alpha-helices and beta-sheets), tertiary (overall 3D structure), and quaternary (arrangement in a protein complex) levels. These intricate 3D structures, dictated by the genetic code, are crucial in determining protein functions and properties. However, proteins are dynamic entities, undergoing conformational changes, interacting with other biomolecules, and being regulated by post-translational modifications, all contributing to their functional complexity.

Among the diverse protein categories, fibrous proteins (Fig. 1.1a), with their long, thin structure, provide mechanical support and structural integrity to cells. Examples include actin-containing microfilaments, microtubules, and intermediate filaments within the cell, and -keratin, the main component of hair, nails, horns, and the epidermis, outside the cell. These proteins are involved in cellular motility, shape determination, and transportation of other molecules.

Enzymes (Fig. 1.1b), another class of proteins, catalyze chemical reactions by converting substrates into products. The cell's chemical reaction network, essential for growth and division, leverages enzymes to overcome energetic barriers and speed up reactions. Enzymes achieve this by stabilizing the transition state of a reaction, significantly lowering the energy required for the reaction to proceed.

The immune system heavily relies on globular proteins called antibodies (Fig. 1.1c). These proteins bind to foreign molecules, or antigens, with high affinity and specificity due to the diversity of the variable region determined by genetic shuffling. The constant region of the antibody interacts with various components of the immune system, determining the subsequent immune response.

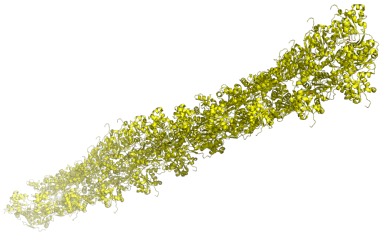
Membrane receptors (Fig. 1.1d), proteins anchored in the cell membrane, enable cells to sense and respond to their environment. They possess an extracellular domain that binds to external molecules (ligands), and an intracellular domain that propagates signals within the cell via signaling pathways. A notable class of membrane receptors is the G protein-coupled receptors (GPCRs), the largest source of drug targets, modulating the activity of various proteins and creating a signaling cascade.

Protein science, with its focus on protein structures, functions, and interactions, remains at the heart of biological research. Given proteins' dynamic nature and functional versatility, computational methods, especially deep learning, offer promising avenues to understand and predict protein behavior.

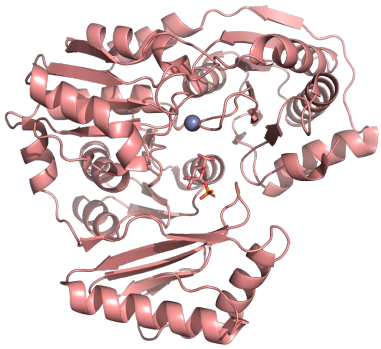
1.2.2 Structure of Proteins: From Amino Acids to Complex Conformations

Proteins are essentially linear polymers composed of fundamental units known as amino acids [107, 56]. There exist 20 distinct amino acids in nature (Fig 1.2), each defined by a unique sidechain, which imparts specific properties to the amino acid. Each amino acid molecule is characterized by a central carbon atom (C_α), linked to a hydrogen atom, an amino group ($\text{-}NH_3^+$), a carboxylate group ($\text{-}COO^-$), and the distinguishing sidechain. These amino acids are connected through peptide bonds, established between the carboxylate group of one amino acid and the amino group of the next. The atoms that are not part of the sidechains collectively form the protein's backbone or mainchain.

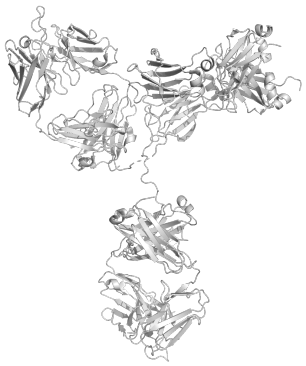
For a protein to fulfill its biological role, it must adopt a specific conformation through a process called protein folding. Protein folding involves a complex dance of atomic interactions, directed by both internal molecular forces and interactions with the surrounding environment, leading to a stable or semi-stable structural state. In biological systems, this folding process occurs either during or immediately after the protein's biosynthesis. The Nobel laureate Christian B. Anfinsen demonstrated in 1973 that, at least for small globular proteins (around 200-300 residues), the protein folding process is entirely governed by its amino acid sequence [8]. Furthermore, he demonstrated that these proteins can regain their functional form, even



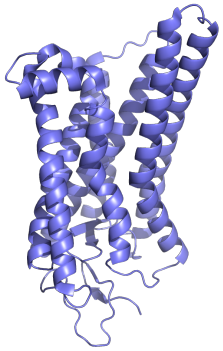
(a) Fibrous actin molecule (PDB: 5MVY)



(b) An enzyme (PDB: 1P5D)



(c) An antibody (PDB: 1HZH)



(d) A GPCR (PDB: 4J4Q)

Figure 1.1: Examples of different protein structures

when unfolded *in vitro*, without assistance from cellular machinery.

Interestingly, protein folding does not occur through exhaustive sampling of all possible conformations, a concept encapsulated in *Levinthal's paradox* [108]. For example, a protein composed of merely 100 residues, each with two potential states, yields a staggering $2^{100} \approx 10^{30}$ potential states. Even if each state transition took only 10^{-13}s, it would still require about 10^{10} years to explore all possibilities - clearly an unrealistic scenario. This apparent paradox has led to the hypothesis that proteins traverse a sequence of intermediary, metastable states, termed the protein's *folding pathway*, before reaching their final conformation. Given the variety of starting conformations possible for an unfolded protein, it is plausible that there exist multiple such pathways for each protein. The *folding funnel* concept captures this idea, suggesting that these distinct pathways are somehow channeled towards a common final state.

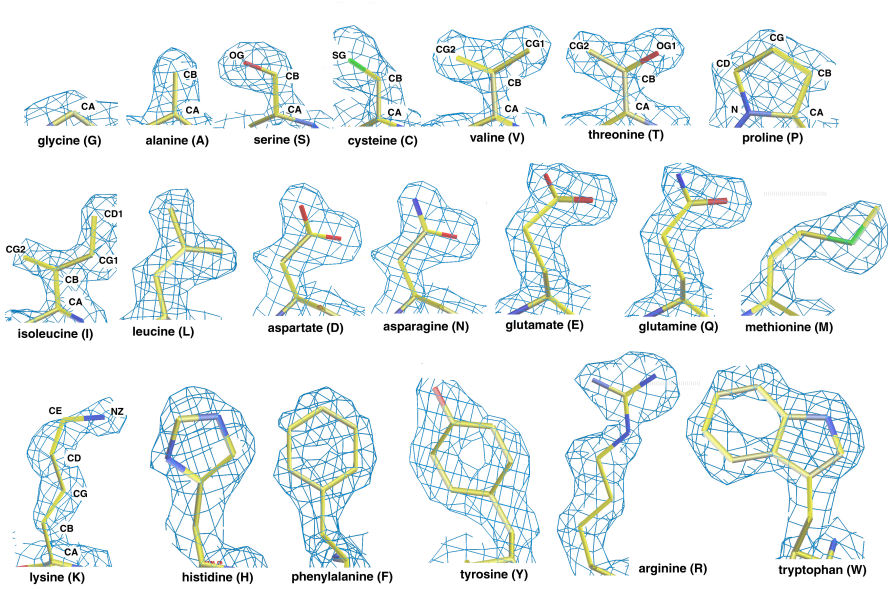


Figure 1.2: The 20 natural amino acids (picture from [145])

The protein's amino acid sequence is referred to as its *primary structure*, while the full three-dimensional conformation is known as its *tertiary structure*. The tertiary structure, in turn, is composed of local three-dimensional segments known as *secondary structure*. Secondary structures arise from hydrogen bonds established between mainchain N-H and C=O groups of proximal residues. The two main types of secondary structures are α-helices and β-sheets (Fig. 1.3), defined by specific bonding patterns and spatial arrangements.

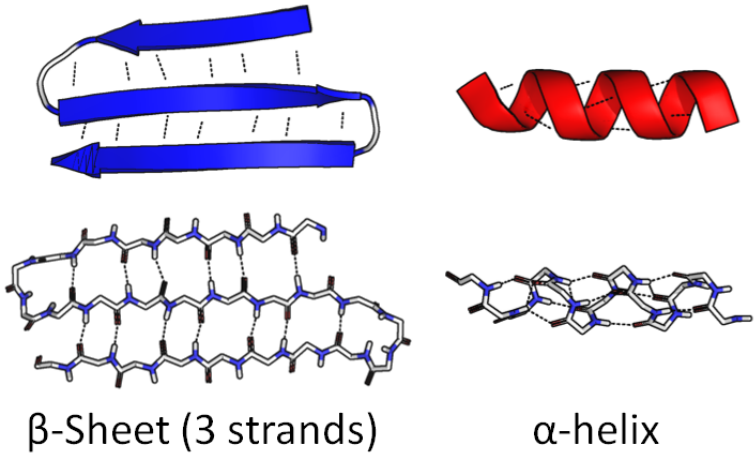
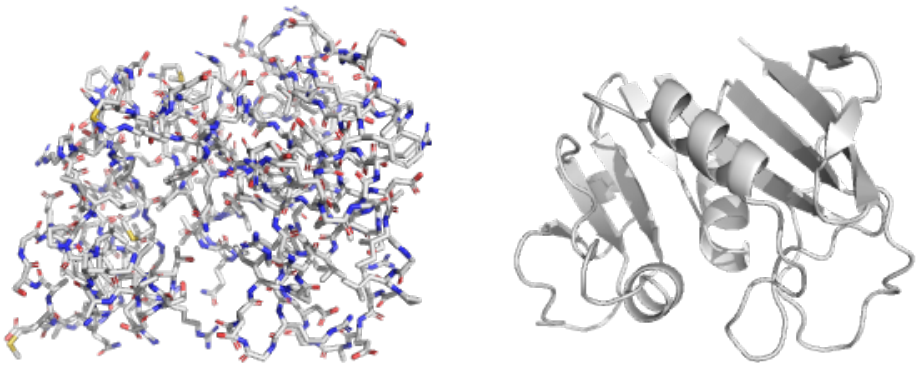


Figure 1.3: Secondary structure elements (picture from [150])

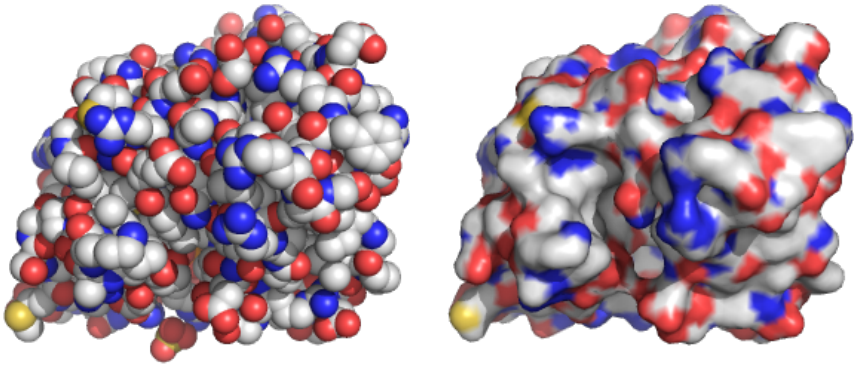
Graphical representation of protein conformation can be achieved through a variety of methods, including stick diagrams, ribbon diagrams, atomic representations, and surface drawings (Fig. 1.4). Each of these visualization methods offers a unique perspective on the protein's structure, providing valuable insight into its biological function. For instance, a ribbon dia-

gram (Fig. 1.4b) accentuates the protein's secondary structures, while a surface (Fig. 1.4d) drawing highlights potential sites of interaction with other molecules. Computational tools, such as PyMOL [148] or UCSF Chimera [125], can generate such graphical representations, and are invaluable resources for researchers seeking to understand protein structure-function relationships.



(a) Stick diagram

(b) Ribbon diagram



(c) Atomic diagram

(d) Surface

Figure 1.4: Different representations of a protein structure

In the context of proteins composed of multiple subunits (a prevalent characteristic in cellular machinery), an additional level of structural organization is recognized as the *quaternary structure*. This term denotes the specific arrangement of subunits, and the interactions between them, in multi-subunit complexes. Understanding the quaternary structure can be crucial in decoding the function of protein complexes, and in guiding drug design efforts that aim to modulate such function.

Protein structure can be decoded using a myriad of methods that can broadly be classified as either experimental or computational [107]. Both of these categories encompass several techniques, each with unique benefits and drawbacks.

In the realm of experimental methods, techniques like x-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and cryogenic electron microscopy (cryo-EM) have been pivotal. Other methods like circular dichroism (CD) spectroscopy and Fourier-transform infrared (FT-IR) spectroscopy provide insights into the broader aspects of protein structure, such as the distribution of various types of secondary structure, even though they do not offer a full atomic resolution picture.

The historical champion in protein structure determination is X-ray crystallography. It exploits the protein's propensity under specific conditions, often at high protein concentration, to assemble into crystalline arrays. The diffraction pattern of an X-ray beam incident on these crystals can then be used to map the electron density within the protein structure, yielding detailed structural insights.

In contrast, NMR spectroscopy sidesteps the need for crystallization, which can be a lengthy and tedious process. It operates by detecting the absorption of electromagnetic radiation by certain NMR-active nuclei (like ^{1}H or ^{13}C) present within the protein structure. The precise absorption frequencies are influenced by the nuclei's local environment, which can help decipher the protein structure. NMR also offers a dynamic snapshot of the protein structure as it collects data from an ensemble of states, providing valuable information about the structure's native environment. However, NMR's application is mainly limited to smaller proteins.

Cryo-EM, a newer entrant in the field of protein structure determination, has the potential to catalyze substantial advances in the field. Cryo-EM involves rapid freezing of a protein solution, followed by the capture of multiple electron microscopy images. As these images provide various angular views of the protein, they can be computationally integrated to construct a comprehensive structural model.

On the computational front, methods for protein structure determination majorly bifurcate into two categories.

The first category includes physics-grounded methods that, using either classical force fields or quantum mechanical approximations like density functional theory (DFT), attempt to discover the protein's most stable, low-energy conformation. These methods explore the conformational landscape in different ways. For instance, molecular dynamics (MD) methods simulate the protein folding process by iteratively computing forces acting on the atoms and adjusting their positions accordingly [146]. Alternatively, methods like Rosetta [141, 103] use a fragment-based approach or homology modeling, incorporating structural patterns from experimentally determined structures to guide their search.

The second category comprises evolutionary-informed methods, which leverage the vast

amounts of DNA sequencing data available. These methods identify residues within the protein that have co-evolved, suggesting that they are likely proximal in the three-dimensional protein structure [44]. The premise is that when one residue mutates, nearby residues are often compelled to mutate as well to accommodate the altered environment.

The advent of machine learning has ushered in a new era of computational methods that aim to directly map from amino acid sequence to protein structure [4, 149, 82], typically taking advantage of co-evolutionary information. These techniques have revolutionized the field of structural prediction and opened up new avenues of research.

1.2.3 Importance of Protein Interactions in Biological Systems

Proteins seldom act alone; their functions are often determined by their interactions with other proteins and biomolecules. Understanding these interactions is therefore crucial for gaining insight into biological processes and systems.

Protein interactions play a critical role in virtually every biological process. They are fundamental for the formation of complex, multi-protein structures such as ribosomes and proteasomes, which are responsible for protein synthesis and degradation, respectively. Protein interactions are also crucial for cellular signaling, wherein signal proteins bind to receptor proteins, triggering a cascade of interactions that transmit signals from the cell surface to the interior.

In metabolism, proteins often function as enzymes that catalyze chemical reactions. These enzymes interact with substrate molecules, facilitating reactions to occur faster or under milder conditions than would be possible otherwise. Moreover, multiple enzymes often interact in metabolic pathways, where the product of one enzyme serves as the substrate for the next.

In the immune system, antibodies (a type of protein) recognize and bind to specific antigens, triggering an immune response. This selective interaction is fundamental for the body's ability to fight off infections.

Proteins also interact with nucleic acids (DNA and RNA), regulating gene expression and playing key roles in DNA replication and repair. For instance, transcription factors are proteins that bind to specific DNA sequences, controlling the transcription of genetic information from DNA to messenger RNA.

Given their central role in biology, protein interactions are crucial in understanding health and disease. Malfunctions in protein interactions can lead to diseases, including cancer, neurodegenerative disorders, and infectious diseases. Thus, understanding protein interactions can inform the development of drugs and therapeutic strategies.

By using computational methods, we can gain insights into these complex protein interactions, opening up opportunities for new discoveries in biology and medicine. This work is part

of such an endeavor, focusing on understanding protein interactions using deep learning techniques.

1.2.4 Forces Governing Protein Folding and Interactions

Understanding the forces that contribute to protein folding and interactions between proteins is crucial to gaining a comprehensive picture of protein function. Predominantly, it is the formation and dissolution of non-covalent bonds that shape these processes. The most prominent among these bonds are van der Waals interactions, hydrogen bonds, ionic bonds, and hydrophobic forces [107, 56, 13].

Van der Waals interactions occur between virtually any pair of atoms that come into proximity. These forces originate from minor fluctuations in the electron clouds of atoms, resulting in temporary dipoles that attract nearby atoms. Remarkably, the strength of these forces varies inversely with the sixth power of the interatomic distance, underscoring the necessity of close contact. However, if atoms approach too closely, a strong repulsive force arises due to the Pauli exclusion principle, which prevents electron cloud overlap.

Hydrogen bonds establish links between a hydrogen atom donor, typically associated with a high electronegativity atom, and an atom acceptor carrying a negative charge. In the watery environment where proteins function, water molecules are a prevalent source of hydrogen bond donors, which significantly influence protein folding and binding.

Ionic bonds, in contrast, form between two groups bearing opposite charges. Many organic molecules, including amino acids with their carboxyl and amino groups, can participate in such interactions.

Hydrophobic interactions, while not technically bonds, significantly contribute to protein folding and interactions. They originate from the disruption of water's hydrogen bonding network caused by the presence of hydrophobic groups. Water molecules tend to form a "shell" around these groups, causing an entropy reduction and free energy increase. As a consequence, hydrophobic groups favor close packing, minimizing their exposure to water.

The properties of the 20 naturally occurring amino acids, notably their interaction with water, are essential in this context. Generally, they are categorized into non-polar, polar, and charged groups. Non-polar or hydrophobic amino acids, like leucine and phenylalanine, often gather in the protein's core, protected from water. Conversely, polar amino acids such as serine can engage in hydrogen bonding with water, thus frequently appearing on the protein's surface. Similarly, charged amino acids, like arginine, also usually occupy the protein's exterior.

This work is particularly focused on protein-protein interactions (PPIs), which play a significant role in most biological functions. The same forces shaping protein folding also drive these interactions. Certain features of protein interfaces, like the buried accessible surface area, are critical. This parameter is computed by deducting the accessible surface area of the

protein complex from that of its individual components. As illustrated by Chen et al. [32], there is a strong correlation between the buried accessible surface area and the dissociation constant (K_d), with larger buried areas resulting in tighter binding. Typically, interfaces bury around 1600 Å^2, and a minimum of 1200 Å^2 is required for stability.

Additionally, the chemical makeup of the interface is influential. Statistical analyses indicate that protein interfaces are chemically intermediate between the hydrophobic protein core and the polar and charged exterior. For effective water exclusion and optimal specificity, interacting interfaces must exhibit both geometrical and chemical complementarity. This encompasses the alignment of hydrogen bonds and charge-charge interactions.

1.3 Introduction to Computational Biology

1.3.1 Role and Importance of Computation in Biology

The field of biology has undergone a transformation in recent decades with the advent of modern computational methods and technologies. Computation has become an indispensable tool across nearly all subdisciplines within biology, enabling researchers to analyze complex biological systems and massive amounts of data in ways not previously possible. At its core, computational biology involves the development and application of mathematical modeling, computational simulation techniques, and data analytics to address biological questions. It allows researchers to integrate diverse datasets, test hypotheses, predict behaviors of biological systems, and identify promising directions for future study. Some of the key areas where computational biology has revolutionized biological research include:

1. **Bioinformatics**: Developing algorithms and methods for analyzing DNA, RNA and protein sequence, structure and function. This includes tasks like sequence alignment, database searching, phylogenetic tree construction, structure prediction, and genomic annotation.
2. **Systems Biology**: Using computational models to study interactions within biological systems and predict systemic behaviors. This provides insights into properties that emerge at the systems-level.
3. **Synthetic Biology**: Redesigning and engineering novel biological systems, such as genetic circuits or metabolic pathways. Computational tools aid in designing circuits.
4. **Pharmacogenomics**: Identifying how genetic variations influence drug responses. Computational methods search for biomarkers to guide personalized medicine.
5. **Population genetics**: Modeling evolutionary dynamics and decoding principles that shape genetic diversity. Simulations examine how mutations spread.
6. **Neuroscience**: Developing computational models of neural processes and principles underlying brain function. Models provide insights hard to obtain via experiments.

7. **Biomedical engineering**: Creating computational models and analytic tools to aid innovations in biomaterials, medical devices, tissue engineering, imaging and diagnostics.

8. **Epidemiology**: Using computational simulations and bioinformatics to analyze disease outbreaks, transmission patterns, and inform public health interventions.

The scale and complexity of biological systems pose immense challenges for traditional experimental methods alone. Computational biology provides an avenue to overcome these hurdles through data-driven modeling and quantitative analysis. As computational power grows and datasets become richer, its role in driving biological discovery will only continue increasing.

1.3.2 Current Computational Methods in Protein Analysis

Proteins are an ideal target for computational methods in biology given their central importance across virtually all biological processes. Some of the key computational techniques used in protein analysis include:

- **Sequence analysis**: Algorithms for searching databases (e.g. BLAST [6]), performing multiple sequence alignments (e.g. ClustalW [160]) and identifying homologous relationships. Provides evolutionary and functional insights.

- **Structure prediction**: Methods for predicting 3D protein structure from sequence using comparative/homology modeling or ab initio simulation (e.g. Rosetta [141], AlphaFold [82]). Allows structure determination when experimental methods fail.

- **Molecular dynamics simulations**: Modeling atomic-level protein dynamics over timescales using physics-based force fields. Reveals stability, flexibility and transient states.

- **Protein design**: Computational redesign of existing proteins or design of novel proteins (e.g. RosettaDesign [42]). Allows engineering proteins with new functions.

- **Protein docking**: Predicting complexes formed between proteins and ligands or other molecules (e.g. AutoDock [119]). Useful for drug design.

- **Interaction prediction**: Identifying potential protein-protein interactions and interaction networks (e.g. STRING [156]). Sheds light on protein function and disease.

- **Function prediction**: Using sequence motifs, structural comparison, machine learning etc. to annotate protein function. Improves characterization of unstudied proteins.

- **Evolutionary analysis**: Phylogenetic approaches for studying protein family evolution (e.g. MEGA [157]). Reveals evolutionary relationships and divergence.

- **Mutation analysis**: Evaluating effect of mutations on protein structure and function using energy-based or machine learning models. Interprets genetic variations.

- **Databases and visualization**: Resources like PDB [20] for 3D structures, UniProt [162] for sequences and Phyre2 [85] for modeling aid computational research.

As experimental techniques continue generating new biological data, computational methods serve as an indispensable partner, providing visualization, analysis and insight extraction. Recent advances in areas like deep learning are opening new frontiers in computational protein research.

1.3.3 An Overview of Protein Design

Protein design represents an exciting scientific field committed to creating proteins with novel or enhanced structures or functions [75]. Broadly, it consists of two main methods: fixed-backbone design and de novo design.

Fixed-backbone design, also referred to as the inverse protein folding problem, requires determining the optimal amino acid sequence that will yield a given protein structure. On the other hand, in de novo design, neither the sequence nor the exact backbone conformation is predetermined. Instead, general guidelines regarding the backbone composition, such as the presence of specific secondary structure elements, are provided. This approach involves computationally sampling various backbone conformations followed by sequence optimization.

Three primary strategies have been employed to design proteins with desired functions. The first, and most conservative, involves modifying the function of an already active protein. For example, this could entail enhancing the binding between two partners by altering the interface residues or adjusting the catalytic function of an enzyme. Typically, such modifications are minor and confined to residues in and near the functional site.

The second approach involves transplanting a functional site from one protein into another, essentially leveraging the structure-function relationship to identify potential host proteins that can accommodate the function within their structure. This process involves recognizing the functional motif in the donor protein and seeking recipient proteins with similar motifs to be adjusted for motif integration. One notable example includes grafting viral epitopes onto smaller scaffold proteins for vaccine design [39].

The third, and most challenging approach, is to design proteins with entirely new functions, such as enzymes with catalytic activities absent in nature or creating novel protein binders [79].

Approaches for designing protein-protein interactions can be organized similarly [147]. Significant progress has been made in the redesign of natural binding sites, particularly through the use of a force function to predict mutations on the interface that boost binding affinity between naturally interacting proteins.

However, the advancement of de novo protein binders has been comparatively slower. This is regrettable given the tremendous potential of designing de novo binders, capable of binding any given molecular surface using scaffold proteins of any desired size, solubility, or stability. Such a capability could revolutionize areas like pharmaceutical development, biosensing, and nanotechnology.

Traditionally the most widely used method for de novo design involves docking and optimization. In this process, naturally occurring proteins are docked against the target surface to identify a scaffold protein that exhibits promising binding characteristics such as high shape complementarity and a large buried surface area. The surface residues of the selected scaffolds are then further optimized for binding using a combination of computational and experimental methods. This approach has yielded some extraordinary designs, like the creation of self-assembling protein complexes on a megadalton scale [14].

The whole field of protein design has in the last few years been shifting away from physics-based methods to find low energy backbones and side chain conformations towards machine learning based methods. ProteinMPNN [43] has for instance been proposed as a replacement for side chain packing and design in Rosetta to do fixed-backbone design. Similarly, in order to go around the problem of painstakingly having to have to specify backbone geometry during de novo design RFDiffusion [169] proposes a score-based generative model of protein backbones. The work presented in this thesis has also resulted in a derived methods for designing de novo protein binders [60].

1.4 Deep Learning and its Role in Computational Biology

1.4.1 Introduction to Deep Learning

Deep learning, an important subclass of machine learning, has made significant strides in a variety of application domains in recent years, including speech recognition, image interpretation, and even geometric deep learning for unstructured data [104, 68, 114]. Rather than relying on manual feature curation and extraction, which often demands a thorough understanding of the problem context, deep learning algorithms are designed to learn relevant features from raw data automatically. In this process, they establish multiple levels of abstraction, creating a hierarchy of learned representations.

The foundation of most deep learning algorithms is the artificial neural network (ANN), which is conceptually inspired by biological neurons in the brain. An individual artificial neuron processes a set of inputs by creating a weighted linear combination of them, adding a bias term, and then applying a non-linear function:

$$y = \sigma(\mathbf{w} \cdot \mathbf{x} + b) \tag{1.1}$$

where $\mathbf{x}$ represents the inputs, y the outputs, $\mathbf{w}$ and b the learnable parameters, and σ a

non-linear function.

A deep neural network (DNN) is essentially an ANN with multiple layers or stages to transform the input into the final output. Each layer in this network is generally made up of multiple neurons. A fully connected layer, for instance, lets each neuron operate on all the inputs:

$$\mathbf{y} = \mathrm{fc}(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1.2}$$

where $\mathbf{W}$ is a matrix and $\mathbf{b}$ is a vector, and together, they encompass all the parameters of the neurons.

A deep fully connected network chains several such layers together: $\mathbf{y} = (\mathrm{fc}_n \circ \ldots \circ \mathrm{fc}_2 \circ \mathrm{fc}_1)(\mathbf{x})$.

In the context of supervised learning, we work with a dataset $D = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1,\ldots,n}$, where each $\mathbf{x}_i$ is an input sample, and $\mathbf{t}_i$ is the corresponding ground truth. The goal is to find the parameters Θ that minimize the difference between the network's outputs and the ground truths. This difference is quantified using a loss or cost function. For instance, in linear regression tasks, we might use the root-mean-square deviation as the loss function: $\mathscr{L}_\Theta(D) = \sum_i \|\mathbf{y}_i - \mathbf{t}_i\|^2$.

Convolutional Neural Networks (CNNs), a specialized type of neural network, have demonstrated excellent performance on Euclidean domains such as images, videos, and acoustic signals [96]. For a given p-dimensional signal on an Euclidean domain $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_p(\mathbf{x}))$, the output of a convolutional layer is $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_q(\mathbf{x}))$, where

$$g_l(\mathbf{x}) = \sigma\left(\sum_{i=1}^{p} (f_i \star \gamma_{l,i})(\mathbf{x})\right) \tag{1.3}$$

and $(f \star \gamma)(\mathbf{x}) = \int_\Omega f(\mathbf{x} - \mathbf{x}')\gamma(\mathbf{x}')\,d\mathbf{x}'$, for a bank of filters $\Gamma = (\gamma_{l,i})$, with $l = 1, \ldots, q$ and $i = 1, \ldots, p$.

For instance, in the case of a digital image, the domain is the (x, y) coordinates of the pixels, and the signal corresponds to the RGB values: $\mathrm{RGB}(x, y) = (R(x, y), G(x, y), B(x, y))$.

The pooling layers of a CNN are defined as $g_l(\mathbf{x}) = P(f_l(\mathbf{x}') : \mathbf{x}' \in \mathscr{N}(\mathbf{x}))$, where $\mathscr{N}(\mathbf{x})$ refers to the neighborhood of $\mathbf{x}$, and P could be a function like the mean or maximum (yielding max-pooling). Pooling layers help to downsample the signal, introducing invariance to minor translations.

CNNs are characterized by three significant properties: sparse interactions, parameter sharing, and translation equivariance. Unlike fully connected layers, where each output unit interacts with all input units, filters in convolutional layers are smaller than the input, leading to sparse interactions. Parameter sharing occurs when the same weights are used as the filter moves across different regions of the input, resulting in translation equivariance. A function is said to exhibit equivariance if the output changes in the same way as the input; thus, if y is equivariant to translation, then $y(\mathscr{T} f) = \mathscr{T} y(f)$, where $\mathscr{T}$ is a translation operator, and f is a signal defined over a domain.

1.4.2 Deep Learning in Bioinformatics: Current Applications and Limitations

Deep learning's ability to automatically learn from large datasets and identify intricate patterns has led to its extensive use in bioinformatics, revolutionizing many subfields within the domain.

In genomics, deep learning has been employed to predict gene expression levels [11], identify genetic variants [135], and annotate genomes [51]. For instance, deep learning models have demonstrated superior performance in predicting the functional effects of genetic variants, which is a critical task for understanding genetic diseases and personalized medicine.

In proteomics, one of the most notable applications of deep learning is the prediction of protein structures. Techniques like AlphaFold [149, 82], which employ deep learning, have achieved remarkable performance in this task, significantly outperforming traditional computational methods.

Deep learning has also found substantial applications in the prediction of protein-protein [59, 54] and protein-ligand interactions [65], which are critical for understanding biological processes and drug discovery. By learning from large-scale interaction datasets, these models can identify interaction sites and predict interaction partners, contributing to our understanding of protein function and the design of new therapeutics.

Despite these advances, applying deep learning to bioinformatics presents unique challenges. Firstly, biological data is often noisy, heterogeneous, and high-dimensional, which can complicate the training of deep learning models. Secondly, many bioinformatics tasks are characterized by a scarcity of labelled data, which is required for supervised learning. Thirdly, the interpretation of deep learning models—a critical aspect in bioinformatics for biological understanding and discovery—is non-trivial due to the "black-box" nature of these models [144].

Additionally, many current models require significant computational resources for training, which may limit their applicability in some settings. Finally, biological data often exhibits complex spatial (e.g., 3D structures of proteins) or temporal (e.g., time-series data from biological processes) patterns, which demand tailored deep learning methods.

This thesis confronts these challenges, developing novel deep learning methods for predicting protein interactions that are computationally efficient, can work with limited labelled data, and provide interpretable predictions, pushing the frontiers of bioinformatics research.

1.4.3 Geometric Deep Learning: An Emerging Tool for Protein Analysis

Geometric deep learning, an umbrella term for various techniques developed to apply the power of Convolutional Neural Networks (CNNs) to non-Euclidean domains such as graphs and manifolds, marks a critical evolution in the field of deep learning [28, 29]. Traditional deep

learning techniques have been primarily developed for Euclidean data such as images and time-series. However, a substantial portion of real-world data, including biological data, exist in non-Euclidean domains, such as graphs and manifolds. Thus, geometric deep learning opens a new avenue for the analysis of these types of data.

One prominent approach in geometric deep learning is the use of Graph Neural Networks (GNNs) [17, 182]. This extension of traditional deep learning techniques to graph data has led to significant advancements in numerous areas, including social network analysis [117], recommendation systems [172], and notably, bioinformatics [177]. In the context of protein analysis, proteins can be represented as graphs where nodes denote amino acids or atoms, and edges signify physical bonds or interactions between them [78]. This approach allows the capture of local and global structural information of proteins, which is crucial for understanding their properties and functions, thereby facilitating various essential tasks in protein analysis.

Moreover, geometric deep learning extends to manifold data, which includes 3D structures of proteins. For manifolds, the concept of *intrinsic convolution* is defined as:

$$(f \star g)(x) = \sum_j g_j D_j(x) f$$

where g_j denotes the filter coefficients, and $D_j(x)$ is the *patch operator* given by:

$$D_j(x) f = \int_\Omega f(x') w_j(x, x') dx'$$

These operators are fundamental for convolutions on manifolds and re-weight the input signal based on the intrinsic properties of the manifold.

Some of the first approaches that were proposed to formulate the weighting functions on manifolds were the Geodesic CNN (GCNN) architecture [113], the Anisotropic Diffusion CNN (ACNN) architecture [26], and the Mixture Model Networks (MoNet) architecture [118]. The first part of this thesis in particular takes advantage of MoNet, which provides the most general construction of the patch operator, with both the GCNN and ACNN operators obtainable as a special configuration of MoNet.

Despite their potential, geometric deep learning methods for protein analysis are still in their infancy, and several challenges remain. These include choosing an appropriate graph or manifold representation for proteins, developing efficient and robust geometric deep learning models, and interpreting these models. This thesis contributes to this burgeoning field by developing novel geometric deep learning methods for predicting protein interactions. We propose new protein representations and model architectures that capture critical interaction fingerprints and demonstrate their effectiveness on several prediction tasks.

1.5 Challenges in Protein Interaction Prediction

1.5.1 Importance and Complexity of Protein Interaction Prediction

Protein interactions play a pivotal role in nearly every biological process. They are the basis of cellular function, enabling complex biochemical cascades, structural formations, and signal transduction. Aberrant protein interactions often lead to pathological conditions, making the accurate prediction and understanding of these interactions crucial not only for elucidating fundamental biological processes but also for therapeutic interventions.

However, the prediction of protein interactions is a complex task. Proteins are dynamic and versatile macromolecules, capable of interacting with various other entities including other proteins, DNA, RNA, and small molecules. These interactions can be transient or stable, specific or nonspecific, and can take place in various cellular environments. They are governed by a multitude of factors, including the physicochemical properties of the interacting entities, their three-dimensional structures, and the cellular context.

From a computational standpoint, the challenge lies in encapsulating these multifaceted characteristics into a predictive model. Traditional methods have relied on features such as sequence information [38], molecular docking [48, 32], or homology-based inference [100]. However, these methods often fall short when it comes to capturing the full complexity of protein interactions. They may not account for the dynamic nature of proteins, may struggle with proteins that have no known homologues, or may require extensive computational resources.

Furthermore, the quality of protein interaction prediction largely depends on the quantity and quality of available interaction data. Experimental techniques for determining protein interactions, such as yeast two-hybrid systems [142] or affinity purification coupled with mass spectrometry [2], are time-consuming, expensive, and often suffer from high rates of false positives and negatives.

In light of these challenges, there is a pressing need for novel computational approaches that can accurately predict protein interactions from readily available data, such as sequence or structure.

1.5.2 Existing Computational Approaches for Interaction Prediction and their Limitations

Over the years, a myriad of computational strategies have emerged to predict protein interactions [74]. Broadly, these can be classified into four categories: homology-based methods, docking simulations, sequence-based methods, and network-based methods. Each category, while offering unique advantages, also presents inherent limitations.

- **Homology-based methods:** Grounded in the premise that proteins with analogous sequences exhibit similar interaction patterns, tools like PSI-BLAST are employed to identify known interaction patterns among homologous proteins [7]. Their major drawback is their diminished efficacy in scenarios with low sequence similarity, rendering them ineffective for proteins with unprecedented folds or functions.

- **Docking simulations:** These methods endeavor to forecast the potential interaction between two proteins in a three-dimensional space by considering the physical and chemical attributes of proteins [138]. However, the computational intensity of docking methods, coupled with the challenges posed by the dynamic nature of protein structures, often limits their accuracy [25]. Furthermore, the precision of these simulations is intrinsically tied to the quality of the input structures.

- **Sequence-based methods:** Focusing exclusively on the amino acid sequences of the involved proteins, techniques such as SVM, Decision Trees, and Random Forests predict interacting pairs [23]. Their limitation lies in their inability to encapsulate the intricate nature of interactions, as they overlook the pivotal three-dimensional structure essential for interaction specificity.

- **Network-based methods:** By analyzing protein-protein interaction networks, these methods leverage the network topology to anticipate new interactions [15]. While potent for system-level interaction analysis, their efficacy is often constrained by the quality and comprehensiveness of interaction networks. Additionally, they seldom offer granular insights into the molecular underpinnings of specific interactions.

General Limitations:

- *Lack of Structural Data:* A significant portion of these methods is contingent upon the presence of high-resolution structural data, which isn't ubiquitously available.

- *Scalability and Computational Costs:* Certain techniques, notably docking, demand substantial computational resources, making them unsuitable for expansive datasets.

- *Handling Dynamic Nature:* The inherent dynamism of proteins, which can alter their conformations during interactions, presents a formidable challenge to account for.

- *False Positives and Negatives:* The veracity of predictions is frequently compromised by elevated rates of false positives and negatives.

- *Lack of Interpretability:* A plethora of machine learning-centric methods yield non-interpretable models, obfuscating the biological rationale behind the predictions.

1.6 Objectives

This dissertation develops a systematic framework for predicting protein-protein interactions using deep learning applied to molecular surfaces. We progressively address key challenges, with each chapter building upon the previous to tackle different facets of this complex prediction task. Specifically, we start by learning interaction fingerprints on fixed surface representations (Chapter 2), then construct surfaces on-the-fly from atomic data (Chapter 3), and finally predict full binding configurations through surface-based docking (Chapter 4). By leveraging surfaces as a unifying representation and integrating deep learning, we are able to predict protein interactions in an end-to-end fashion. Overall, this dissertation seeks the development of protein interaction prediction methods that incorporate deep learning and data-driven techniques. The objectives are two-fold: (1) analyze protein surfaces to precisely characterize their interaction fingerprints, and (2) enable the accurate prediction of binding partners and complexes based on these fingerprints. The methods utilize advanced machine learning techniques to automate prediction and propose innovative solutions to limitations of existing approaches.

1.6.1 Aim 1: Learning Interaction Fingerprints on Molecular Surfaces

Molecular surfaces provide a higher-level abstraction of protein structure, capturing geometric and chemical patterns indicative of interactions. In Chapter 2, we present MaSIF, a conceptual framework leveraging geometric deep learning to decipher interaction fingerprints on protein surfaces. MaSIF transforms surface patches into numerical descriptors using neural networks applied in geodesic space. We demonstrate MaSIF's versatility across three distinct prediction tasks:

1) Classifying ligand binding pockets based on local surface patterns. MaSIF achieves state-of-the-art accuracy in distinguishing binding sites preferences.

2) Identifying protein-protein interaction sites by learning global surface features. MaSIF convincingly labels interface regions, outperforming other methods.

3) Ultrafast scanning of surfaces to predict binding partners. By encoding complementarity, MaSIF enables rapid search for potential interactors.

Through these applications, we showcase MaSIF's ability to discover interaction fingerprints on diverse protein surfaces solely from geometric and chemical properties.

1.6.2 Aim 2: End-to-End Learning from Atomic Coordinates

While powerful, MaSIF relies on hand-crafted surface representations and input features. In Chapter 3, we introduce dMaSIF which constructs surface representations directly from protein atomic coordinates and learns all features in an end-to-end manner. dMaSIF gener-

ates surface point clouds on-the-fly and computes chemical properties using small neural networks. We also implement a novel convolutional operator that establishes quasi-geodesic neighborhoods by approximating geodesic distances using point clouds.

Remarkably, dMaSIF achieves comparable accuracy to MaSIF on binding site prediction and protein-protein interaction tasks, while improving efficiency by an order of magnitude. By enabling end-to-end learning, dMaSIF opens the door to fully differentiable optimization in generative modeling tasks such as protein design.

1.6.3 Aim 3: Enhancing Protein-Protein Docking with Surface-Based Representations

Chapter 4 delves into the challenge of predicting protein-protein complexes, introducing DiffMaSIF, a novel score-based diffusion model for rigid protein-protein docking. Unlike prior ML methods that relied on residue representations, DiffMaSIF employs a surface-based molecular representation, capturing the essential complementarity of protein interfaces. We also address structural leakage in a popular training dataset and establish new benchmarking splits. The results underscore DiffMaSIF's advantages over contemporary ML methods and its comparable performance to traditional docking tools, all while generating significantly fewer decoys.

2 Deciphering interaction fingerprints from protein molecular surfaces

This chapter is a postprint version based on an article published in Nature Methods in 2020 (DOI: 10.1038/s41592-019-0666-6) in accordance with the publisher.

Authors
Pablo Gainza[1,2], **Freyr Sverrisson**[1,2], Frederico Monti[3,4], Emanuele Rodolà[5], Davide Boscaini[6], Michael M. Bronstein[3,4,7] & Bruno E. Correia[1,2]

Affiliations
[1] Institute of Bioengineering, École Polytechnique Fédérale de Lausanne, [2] Swiss Institute of Bioinformatics, Lausanne, Switzerland, [3] Institute of Computational Science, Faculty of Informatics, USI, Lugano, Switzerland, [4] Twitter, London, UK, [5] Department of Computer Science, Sapienza University of Rome, Rome, Italy, [6] Technologies of Vision Unit, Fondazione Bruno Kessler, Trento, Italy, [7] Department of Computing, Imperial College London, London, UK.

Author contributions
P.G., F.S., F.M., M.M.B. and B.E.C designed the overall method and approach. M.M.B. and B.E.C supervised the research. P.G., F.M. and F.S. developed the base MaSIF method. P.G. designed and implemented MaSIF-site and MaSIF-search. F.S. designed and implemented MaSIF-ligand. F.S. and P.G. developed MaSIF-search's second-stage alignment algorithm. F.S. and P.G. developed the second-stage scoring neural network. P.G., F.S., M.M.B. and B.E.C. analyzed the data. E.R. and D.B. assisted in the design and development of these methods. P.G., F.S., M.M.B. and B.E.C wrote the manuscript. All authors read and commented the manuscript.

2.1 Abstract

Predicting interactions between proteins and other biomolecules solely based on structure remains a challenge in biology. A high-level representation of protein structure, the molecular surface, displays patterns of chemical and geometric features that fingerprint a protein's modes of interactions with other biomolecules.

We hypothesize that proteins participating in similar interactions may share common fingerprints, independent of their evolutionary history. Fingerprints may be difficult to grasp by visual analysis but could be learned from large-scale datasets. We present MaSIF (Molecular Surface Interaction Fingerprinting), a conceptual framework based on a geometric deep learning method to capture fingerprints that are important for specific biomolecular interactions.

We showcase MaSIF with three prediction challenges: protein pocket-ligand prediction, protein-protein interaction site prediction, and ultrafast scanning of protein surfaces for prediction of protein-protein complexes. We anticipate that our conceptual framework will lead to improvements in our understanding of protein function and design.

2.2 Main

Interactions between proteins and other biomolecules are the basis of protein function in most biological processes. Predicting these interactions purely from structure remains one of the most important challenges in structural biology [46, 180, 73, 94]. Many programs effectively predict these interactions by exploiting evolutionary signatures in protein sequence and structure [176, 129, 38], yet these approaches require the knowledge of homologous proteins. The molecular surface [136] is a higher-level representation of protein structure that models a protein as a continuous shape with geometric and chemical features. We propose that molecular surfaces are fingerprinted with patterns of chemical and geometric features that reveal information about the protein's interactions with other biomolecules. Our central hypothesis is that proteins with no sequence homology that undergo similar biomolecular interactions may display similar patterns, which are difficult to grasp by visual analysis but could be learned from large-scale datasets. Here, we present MaSIF (Molecular Surface Interaction Fingerprinting), a general geometric deep learning [28] method to recognize and decipher patterns on protein surfaces, without explicit consideration of the underlying protein sequence or structural fold.

The molecular surface representation describing protein structure (Fig. 2.1a) has long been used for many tasks involving protein interactions [152, 48], and has been the preferred structural description to study protein:solvent electrostatic interactions [151]. More recently, several methods have captured molecular surface patterns with functional relevance, such as 3D Zernike descriptors [41, 88, 183, 164] and geometric invariant fingerprint descriptors (GIF) [178]. These approaches proposed 'handcrafted' descriptors, manually-optimized vectors which describe protein surface features. The scope of these approaches is limited as it is hard

to determine a priori the right set of features for a given prediction task.

Geometric deep learning [28] (GDL) is a nascent field extending successful image-based deep neural network architectures, such as convolutional neural networks [97], to geometric data such as surfaces, where these techniques have been shown to significantly outperform handcrafted feature extraction [118, 113]. MaSIF exploits GDL to learn interaction fingerprints in protein molecular surfaces. The molecular surface data is described in geodesic space, meaning that the distance between two points corresponds to the distance of 'walking' between the points along the surface. In highly irregular protein surfaces (e.g. with deep pockets), geodesic distances can be much larger than Euclidean distances (Supp. Fig. 2.6). First, MaSIF decomposes a surface into overlapping radial patches with a fixed geodesic radius (Fig. 2.1a-b). Each point within a patch is assigned an array of geometric and chemical input features (Fig. 2.1b). The input features (chemistry and geometry) are not learned, they are pre-computed properties from the molecular surface. MaSIF then learns to embed the surface patch's input features into a numerical vector descriptor (Fig. 2.1d). Each descriptor is further processed with application-dependent neural network layers. The networks are trained end-to-end, meaning that the intermediate patch descriptors are not universal but rather optimized towards particular tasks.

We showcase MaSIF with three proof-of-concept applications (Fig. 2.1e): a) ligand pocket similarity comparison (MaSIF-ligand); b) protein-protein interaction (PPI) site prediction in protein surfaces (MaSIF-site); c) ultrafast scanning of surfaces, where we exploit surface fingerprints to predict the structural configuration of protein-protein complexes (MaSIF-search). Our conceptual framework will be useful for biologists that search for similar interaction fingerprints between proteins with no shared evolutionary ancestry. Crucially, MaSIF represents a departure from learning on Euclidean structural representation and may enable the recognition of important structural features for protein function and design.

2.3 MaSIF - A general framework to learn protein surface fingerprints

The MaSIF conceptual framework is shown in Fig. 2.1 and described in the Methods section. Briefly, from a protein structure we compute a discretized molecular surface (solvent excluded surface) [143] and assign geometric and chemical features to every point (vertex) in the mesh (Fig. 2.1a-b). Around each vertex of the mesh, we extract a patch with geodesic radius of r=9 Å or r=12 Å (Fig. 2.1b). The choice of patch radius is application-dependent, in architectures with multiple geodesic convolutional layers we use smaller patch size due to memory limitations (see Methods). For each vertex within the patch, we compute two geometric features (shape index [91] and distance-dependent curvature [178]) and three chemical features (hydropathy index [99], continuum electrostatics [83], and the location of free electrons and proton donors [93]). The vertices within a patch are assigned geodesic polar coordinates (Fig. 2.1c): the radial coordinate, representing the geodesic distance to the center of the patch; and the angular

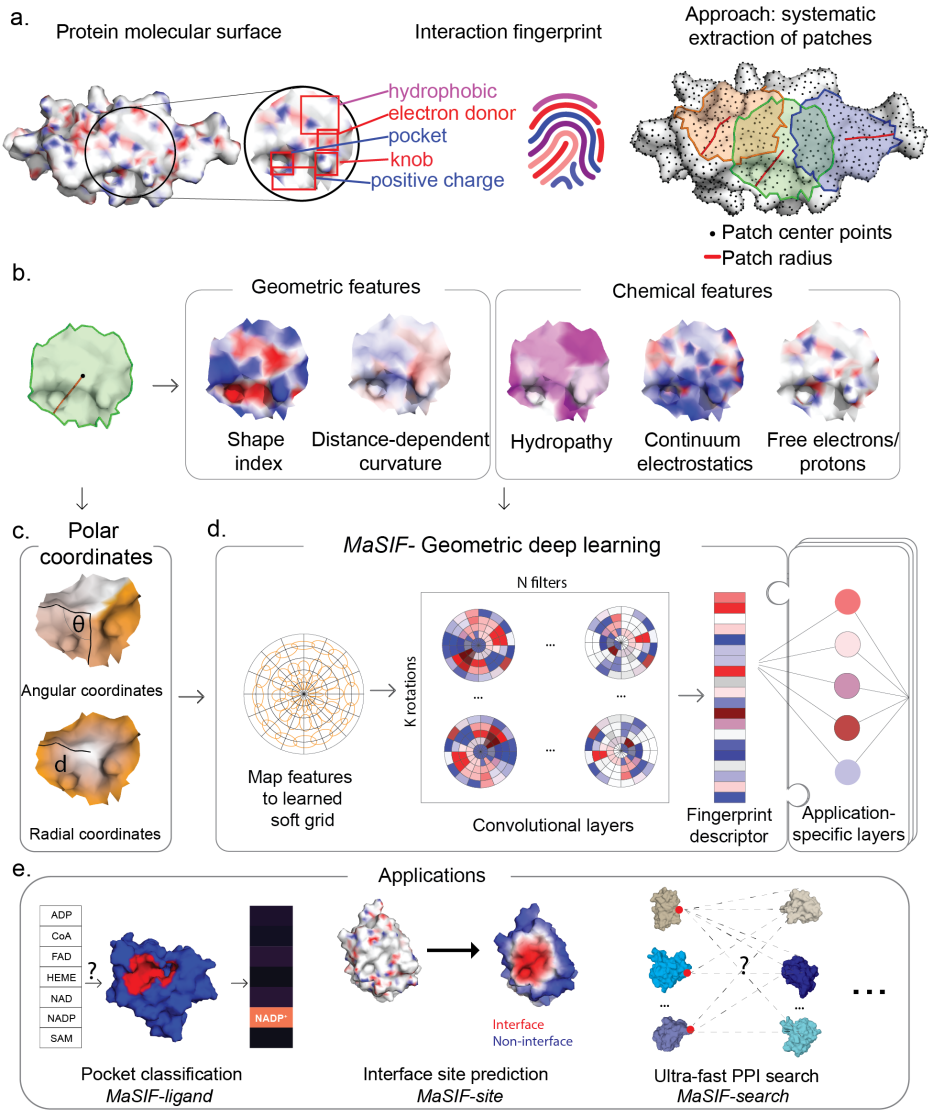


Figure 2.1: Overview of the MaSIF conceptual framework, implementation, and applications. a. Left, conceptual representation of a protein surface engraved with an interaction fingerprint, surface features that may reveal their potential biomolecular interactions. Right, surface segmentation into overlapping radial patches of a fixed geodesic radius used in MaSIF. b. The patches comprise geometric and chemical features mapped on the protein surface. c. Polar geodesic coordinates used to map the position of the features within the patch. d. MaSIF uses geometric deep learning tools to apply convolutional neural networks to the data. Fingerprint descriptors are computed for each patch using application-specific neural network architectures, which contain reusable building blocks (geodesic convolutional layers). e. MaSIF is generalizable and applicable to multiple prediction tasks - a selected few are showcased in this paper.

coordinate, computed with respect to a random direction from the center of the patch, as the patch lacks a canonical orientation. The geometric structure of the surface (e.g. the 'depth' of

a pocket within the surface) are implicitly described through the geometric features (shape index and distance-dependent curvature) and the geodesic polar coordinates.

MaSIF applies a geometric deep neural network to these input features using the polar coordinates to spatially localize features. The neural network consists of one or more layers applied sequentially; a key component of the architecture is the geodesic convolution, generalizing the classical convolution to surfaces and implemented as an operation on local patches [113]. In the polar coordinates, we construct a system of Gaussian kernels defined in a local geodesic polar system for which the parameters are learnable. The learnable Gaussian kernels locally average the vertex-wise patch features (acting as soft pixels) and produce an output of fixed dimension, which is correlated with a set of learnable filters [118]. We refer to this family of learnable Gaussian kernels as a learned soft polar grid (see Methods).

A convolutional layer with a set of filters is then applied to the output of the soft polar grid layer. Note that since the angular coordinates were computed with respect to a random direction, it becomes essential to compute information that is invariant to different directions (rotation invariance, Fig. 2.1d). To this end, we perform K rotations on the patch and compute the maximum over all rotations [113], producing the geodesic convolution output for the patch location. The procedure is repeated for different patch locations similar to a sliding window operation on images, producing the surface fingerprint descriptor at each point, in the form of a vector that embeds information about the surface patterns of the center point and its neighborhood. The learning procedure consists of minimizing the parameter set of the local kernels and filter weights with respect to the application-specific training data and cost function. Therefore, the parameter set is specific to each application presented here.

With this framework we created descriptors for surface patches that can be further processed in neural network architectures. Next, we will present various ways to leverage them to identify interaction fingerprints on protein surfaces.

2.4 Results

2.4.1 Molecular surface fingerprinting to classify ligand binding pockets

Interactions between proteins and metabolites play a fundamental role in cellular homeostasis, yet our knowledge of these interactions is extremely limited [36]. We propose that the interaction fingerprints in protein surfaces hold information to decipher the metabolite-binding preference of protein pockets. To test this hypothesis, we developed MaSIF-ligand, a classifier to predict the metabolite binding preference of a pocket from surface features (Fig. 2.2a). For this proof-of-concept we used seven cofactors: ADP, NAD, NADP, FAD, SAM, CoA and HEME, metabolites with large structural datasets available (Fig. 2.2b).

We trained MaSIF-ligand on a large set of cofactor-binding proteins using their holo-structures, where sequences and structures were clustered to remove redundancy from the training and

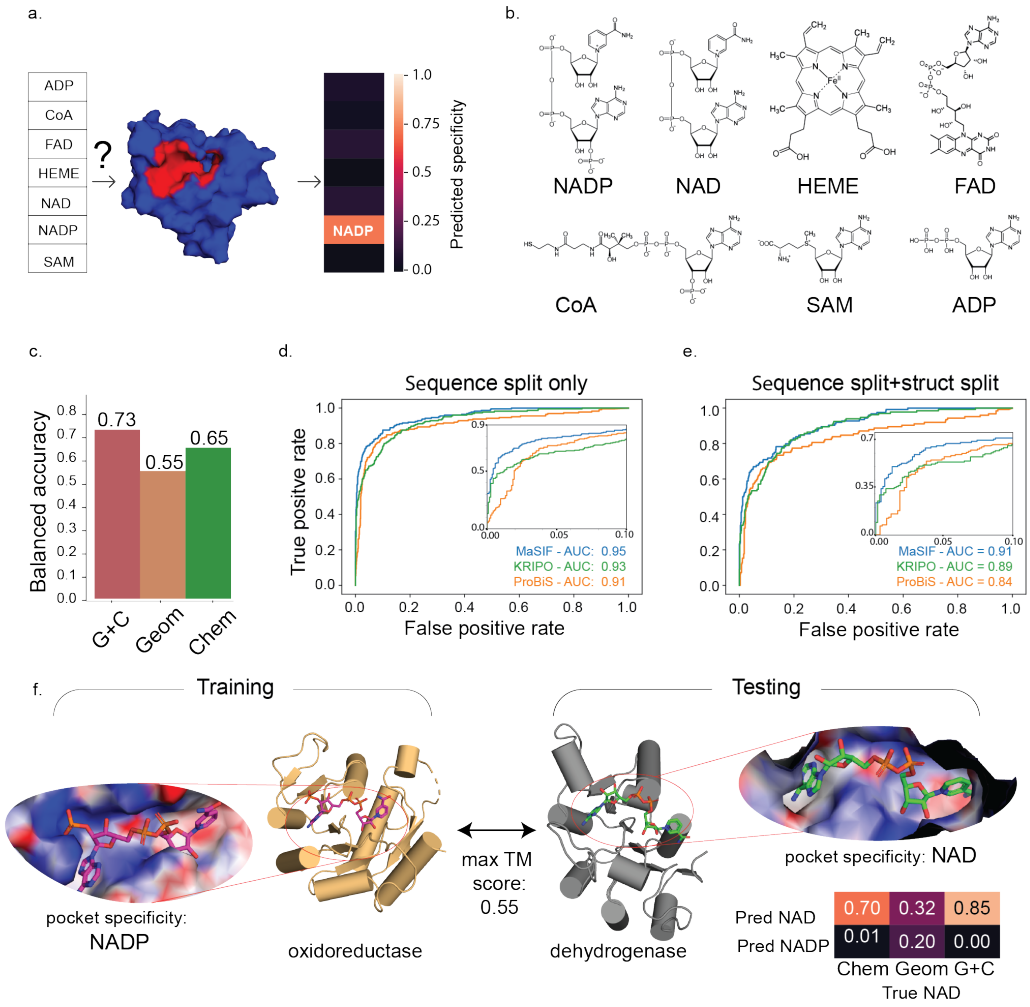


Figure 2.2: Classification of ligand binding sites using MaSiF-ligand. a. Schematic representation of the prediction task. The neural network receives a protein pocket as input and classifies it into seven categories to reflect the predicted binding preference. b. Structures of the seven cofactors that bind proteins considered for the prediction task. c. Balanced accuracy of the prediction of the specificity of binding sites using all features (G+C - Geometry and Chemistry), only geometric features (Geom), or only chemical features (Chem) d. ROC curves for comparative benchmarks for pocket classification using the full training and testing sets (excluding HEME, total number of pockets in testing set was 216). e. ROC curves for comparative benchmarks using a strict structural split of the pockets between the training and test sets (TM score < 0.5, total number of pockets in testing set was 121) f. Specific example on a protein fold that recognizes two similar ligands and yet is correctly predicted. A bacterial dehydrogenase in the test set binds to NAD (PDB id: 2O4C) [70], while its closest structural homologue in the training set corresponds to a mammalian oxidoreductase (PDB id: 2YJZ), which binds to NADP [66].

test sets. The balanced accuracy on an independent test set was used to gauge the classification power of MaSIF-ligand. We first trained MaSIF-ligand with all features (geometry and chemistry) and obtained a balanced accuracy of 0.73 (Fig. 2.2c) (expected random accuracy: 0.14). To investigate the importance of the features, we limited the set to geometric or chemical features which reduced the balanced accuracy to 0.55 and 0.65, respectively (Fig. 2.2c).

Next, we compared MaSIF-ligand with three other programs, ProBiS [92], KRIPO [139], and SiteEngine [152], which exploit structural features for pocket classification, and showed top-tier performance in a recent comprehensive benchmark [50]. To compare the different methods we use the Receiver Operator Characteristic Area Under the Curve (ROC AUC). In our datasets, SiteEngine is the top performer among these tools, while MaSIF-ligand achieves a better performance than KRIPO and ProBiS (Fig 2d). Both SiteEngine and MaSIF-ligand identify physicochemical and geometric similarities in molecular surfaces. However, SiteEngine is based on explicit alignments of pockets using pseudo-representations of the molecular surface, which results in a much higher runtime. It is therefore remarkable that MaSIF-ligand can achieve similar performances despite embedding the 3D-space into fingerprint descriptors.

To analyze the MaSIF-ligand predictions in detail, we generated a confusion matrix with all features (Supp. Fig. 2.7a). We observe variable performances across ligands, perhaps not surprisingly in the case of HEME (accuracy of 94%) given the chemically dissimilarity to the other cofactors. More challenging is the distinction between similar ligands, namely in the analysis of the confusion data between two highly similar cofactors: SAM vs. ADP and NADP vs. NAD. In both cases, the geometric features are not sufficient and are mainly the chemical features that contribute to the correct predictions (Supp. Fig. 2.7a-b). The capacity of MaSIF-ligand to distinguish the features from very similar cofactors is remarkable, especially for NADP vs. NAD, which differ by a single phosphate group on the adenosine moiety. To understand these successful predictions, we analyzed the pocket features of an NAD-binding bacterial dehydrogenase [70] in our test set and its closest structural homologue in the training set, a mammalian oxidoreductase which binds to NADP (Fig. 2.2f) [66]. We analyzed the regions of the pockets giving the neural network the highest discrimination score between NAD vs. NADP, and mapped this score on the pocket surface (see Methods) (Supp. Fig. 2.7c). The largest discrimination scores arise from patches centered around the additional NADP phosphate in the oxidoreductase:NADP pocket, while in the dehydrogenase:NAD pocket, the adenine moiety region, where NAD and NADP differ, is crucial to correctly classify the pocket. The prediction probabilities for the dehydrogenase:NAD pocket are dependent on the chemical features (Fig. 2.2f, right), further confirmed by the Poisson-Boltzmann electrostatics showing that the oxidoreductase:NADP pocket (Fig. 2.2f, left) has a stronger positive charge distribution, consistent with its binding to the more negatively charged NADP.

Despite the lack of global sequence homology and structural similarity of the pockets in the test and training sets, MaSIF-ligand can decipher the surface interaction fingerprints to determine the binding preference of each pocket. As illustrated by the NAD/NADP example MaSIF-ligand can infer the correct cofactor in two proteins with the same fold based purely on surface

features, without explicit consideration of the underlying amino acids or sequence-based signatures.

Overall, the interaction fingerprints in protein surfaces could be an additional source of information available to biologists to infer important protein:ligand interactions.

2.4.2 Predicting protein binding sites based on interaction fingerprints

Inspired by previous work on PPI site prediction [80, 130, 123], we developed MaSIF-site, a classifier that receives a protein surface as input and outputs a predicted score for each surface vertex on the likelihood of being involved in a PPI (Fig. 2.3a.).

MaSIF-site was trained and tested on a large dataset of protein structures that were co-crystallized in the holo state and separated into monomeric subunits. The training and testing sets were split based on sequence and structure (see Methods). This task greatly leverages the potential of deep learning approaches, since multiple layers yield superior predictions (Fig. 2.3b). Using one geodesic convolutional layer MaSIF-site's ROC AUC reaches 0.77 (Fig. 2.3b and Supp. Fig. 2.8), while three layers boost the ROC AUC to 0.86, computed over all the surface points of the test set proteins.

A strong separation between the predicted true and false interfaces is observed (Fig. 2.3c). A feature ablation study showed that the Poisson Boltzmann continuum electrostatics (Elec) reached the highest performance (ROC AUC=0.80) of all single feature (Fig. 2.3d), suggesting an important contribution of electrostatics on the identification of PPI sites.

Surfaces involved in PPIs can be classified according to biophysical (e.g. obligate vs transient) and structural/chemical (e.g. large vs small, hydrophobic vs polar, etc.) properties, we asked whether MaSIF-site had a biased performance for a particular type of surface (Fig. 2.3e). These predictions were reported in median ROC AUC per protein providing a better assessment of the performance for each query protein. The prediction accuracy for the whole dataset reached a median ROC AUC of 0.87 per protein, while for a subset of transient interactions the ROC AUC was 0.81. Proteins with large hydrophobic interfaces had a better performance (ROC AUC=0.89) than those with the smallest hydrophobic surfaces (ROC AUC=0.81). The median ROC AUC value is illustrated with the example of Ubiquitin Hydrolase (ROC AUC=0.84), close to the median of the whole dataset (Fig. 2.3f).

We compared MaSIF-site to top performing predictors SPPIDER [130] and PSIVER [121], in a subset of transient interactions which are likely amongst the most challenging test cases. MaSIF-site reaches the highest performance, median ROC AUC per protein of 0.81, while SPPIDER and PSIVER reach 0.65 and 0.62, respectively (Fig. 2.3g). The distribution of ROC AUCs per protein for each method is shown in Supp. Fig. 2.9b. We further illustrate MaSIF's superior performances relative to SPPIDER in four randomly chosen proteins from the transient test set (Supp. Fig. 2.9c).

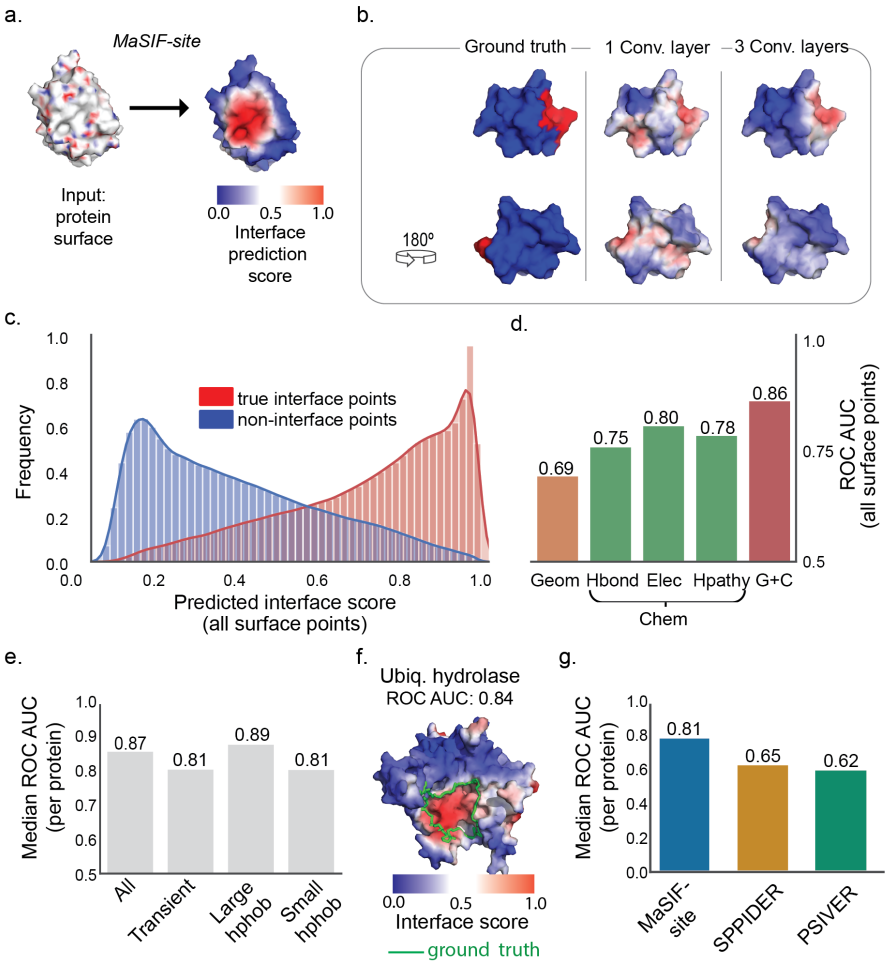


Figure 2.3: Prediction of surface patches involved in PPIs. a. Schematic representation of the interface site prediction workflow. The MaSIF-site receives as input a protein surface with a descriptor vector and outputs a surface score that reflects the predicted interface propensity (red for high interface propensity, blue for low propensity). b. Visual comparison between MaSIF-site with a network with 1 convolutional layer vs. 3 convolutional layers. c. Distribution of predicted scores for true positives (red) vs. true negatives (blue) for a network trained with all features. The ROC AUC values were computed based on the surface points in the proteins of the test set. d. ROC AUC scores for ablation studies with networks trained with different subsets of features: only geometric (Geom), only the location of free electrons/proton donors (hbond), Poisson-Boltzmann electrostatics (elec), the hydropathy index (hpathy), and all features (G+C) (surface points: # positives=218246, # of negatives=1973624). e. Left: Median ROC AUCs (per protein) for selected subsets of proteins. All - full test set containing all proteins (361 proteins); Transient - proteins forming known transient interactions (59 proteins); Large hphob - protein complexes with interfaces composed of mostly hydrophobic residues (74 proteins); Small hphob - protein complexes with small hydrophobic interfaces (74 proteins). f. An illustrative example of a protein with a ROC AUC close to the median of 0.84, which is close to the median of MaSIF-site. g. Comparison of MaSIF-site with the SPPIDER and PSIVER predictor for a set of 53 single-chain transient interactions. Results are shown as the median ROC AUC per protein, evaluated on a per-residue basis for comparison with the other predictors.

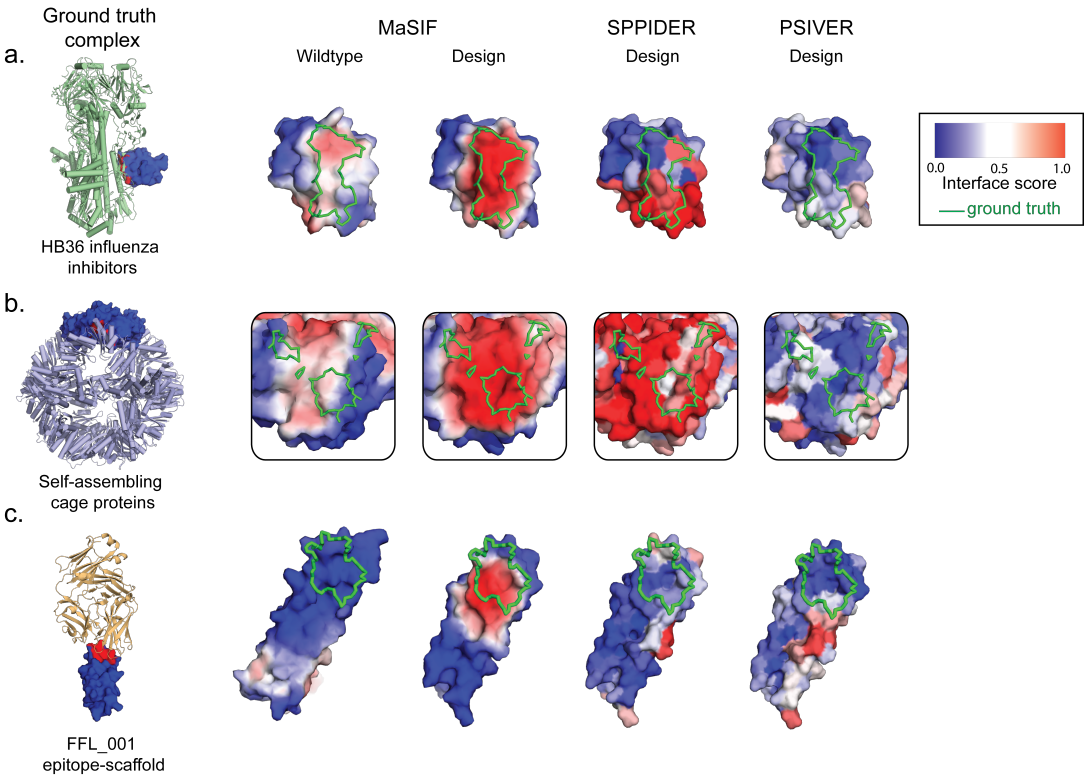


Figure 2.4: Prediction of PPI sites on a set of computationally designed proteins. a. Designed HB36 influenza inhibitors (PDB id: 3R2X) vs. the wild type scaffold protein (PDB id: 1U84). b. Designed self-assembling nanocage protein (PDB id: 3VCD) vs. the wild type scaffold (PDB id: 3N79) c. Designed Respiratory syncytial virus epitope-scaffold (PDB id: 4JLR) vs. wild type scaffold (PDB id: 1ISE). MaSIF-site was tested on a set of de novo computationally-designed proteins involved in PPIs, where the prediction on the designed binders was compared to the corresponding native proteins. For comparison, predictions with SPPIDER and PSIVER were generated for the designed proteins (right).

Although evolutionary information can be crucial to predict protein interaction sites [121], in some cases such evolutionary history is sparse or completely absent. These extreme cases include computationally designed PPIs, whose interfaces were rationally designed in protein scaffolds. We used MaSIF-site to predict three such designed interfaces that have been experimentally validated: an influenza inhibitor [57] (Fig. 2.4a), a homo-oligomeric cage protein [89] (Fig. 2.4b), and an epitope-scaffold used as an immunogen [39] (Fig. 2.4c). The designs were based on wildtype scaffold proteins with no binding activity, and in each case, we compared their interface score with that of the non-interacting wildtype. MaSIF-site clearly labels the interfaces of the designs, in contrast with SPPIDER and PSIVER's predictions. Overall, MaSIF-site may help to identify the sites of interactions with other proteins for PPI validation, paratope/epitope prediction, or small molecule binding sites, for cases where evolutionary or experimental information may not be available.

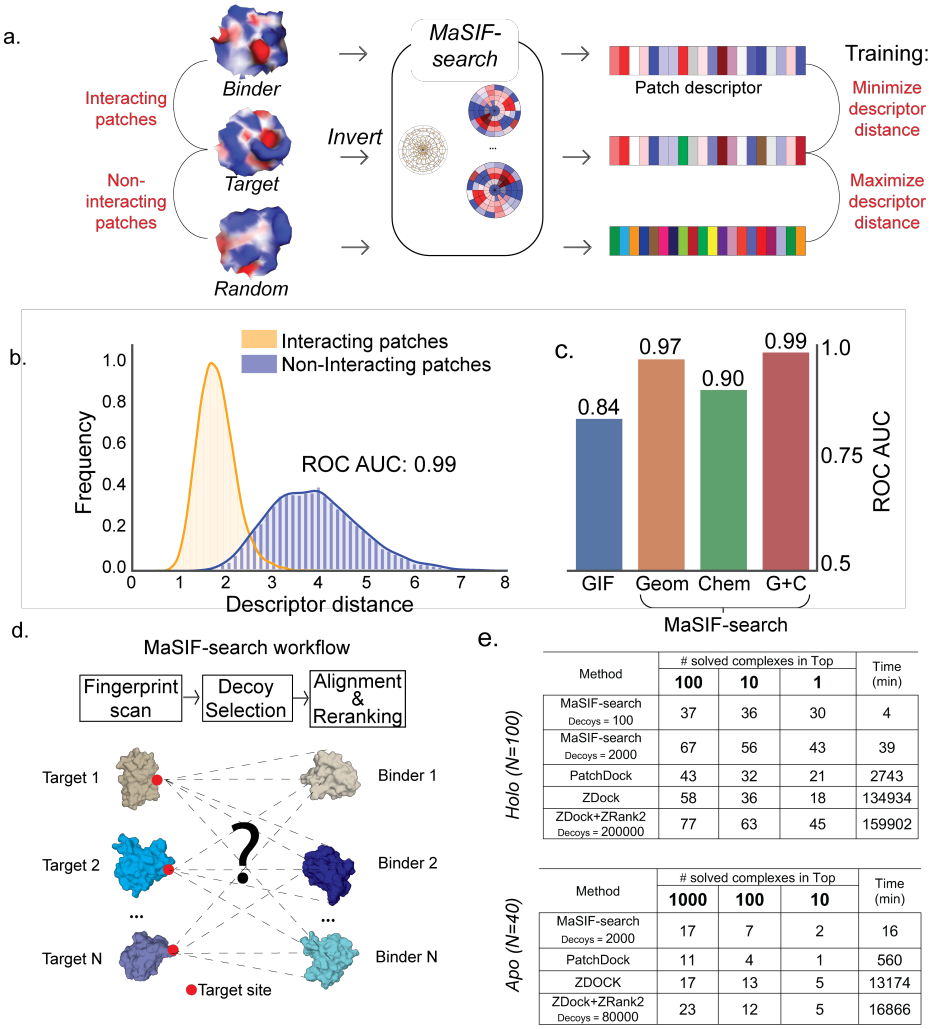
2.4.3 Ultrafast scanning of interaction fingerprints for prediction of protein-protein complexes

As a last example of MaSIF's generality, we show the embedding of fingerprints as vectorized descriptors to predict specific interactions between proteins. This embedding, inspired by earlier work on GIF descriptors [178], is attractive because, once the descriptors are precomputed, nearest-neighbor techniques can scan billions of descriptors per second [120]. The gain in computational cost at runtime enables broad structural searches across large databases, moving away from the paradigm of 1 binder vs. 1 target, typical of docking programs, to one of many binders vs. many targets. This is important for tasks such as protein design, where docking tools are used to search for structural templates to use as starting points for the design of novel PPIs or ligand-binding proteins [57]. Thus, we introduce MaSIF-search, a new paradigm for the fast search of protein binding partners based on surface fingerprints. MaSIF-search is then complemented with surface alignment and reranking stages to generate docked complexes with improved quality.

MaSIF-search learns patterns in interacting pairs of surface patches. PPIs occur through surface patches with some degree of complementary geometric and chemical features. To formalize this observation, MaSIF-search inverts the numerical features of one protein partner (multiplied by -1), with the exception of hydropathy. Although the models of complementarity are not perfect the network may be able to learn different levels of complementarity. After performing the inversion on one patch, the Euclidean distance between the fingerprint descriptors of two complementary surface patches should be close to 0. Within this framework, MaSIF-search will produce similar descriptors for pairs of interacting patches (low Euclidean distances between fingerprint descriptors), and dissimilar descriptors for non-interacting patches (larger Euclidean distances between fingerprint descriptors) (Fig. 5a). Thus, identifying potential binding partners is reduced to a comparison of numerical vectors.

To test this concept, we assembled a database with >100K pairs of interacting protein surface patches with high shape complementarity, as well as a set of randomly chosen surface patches, to be used as non-interacting patches. A trio of protein surface patches with the labels, binder, target, and random patches were fed into the MaSIF-search network (Fig. 5a). The neural network is trained to simultaneously minimize the Euclidean distance between the fingerprint descriptors of binders vs targets, while maximizing the Euclidean distance between targets vs random, commonly referred to as a Siamese architecture in the machine learning literature [35] (see Methods).

Performance on the test set shows that the descriptor Euclidean distances for interacting surface patches is much lower than that of non-interacting patches, resulting in a ROC AUC of 0.99 (Fig. 5b). Our method is directly comparable to the previously proposed handcrafted GIF descriptors [178], which were proposed for a similar application: screening functional protein surfaces. Tested on our test set, GIF descriptors show a ROC AUC of 0.84, significantly lower than MaSIF-search (Fig. 5c). Testing MaSIF-search using only chemical or geometric features,



Holo (N=100)

Method	# solved complexes in Top 100	10	1	Time (min)
MaSIF-search Decoys = 100	37	36	30	4
MaSIF-search Decoys = 2000	67	56	43	39
PatchDock	43	32	21	2743
ZDock	58	36	18	134934
ZDock+ZRank2 Decoys = 200000	77	63	45	159902

Apo (N=40)

Method	# solved complexes in Top 1000	100	10	Time (min)
MaSIF-search Decoys = 2000	17	7	2	16
PatchDock	11	4	1	560
ZDOCK	17	13	5	13174
ZDock+ZRank2 Decoys = 80000	23	12	5	16866

Figure 2.5: Prediction of PPIs based on surface fingerprints. a. Overview of the MaSIF-search neural network optimization (Siamese architecture) to output fingerprint descriptors, such that the descriptors of interacting patches are similar, while those of non-interacting patches are dissimilar. The features of the target patch (with the exception of the hydropathy features) are inverted to enable the minimization of the fingerprint distance. b. Distribution of fingerprint distances showing interacting (yellow) and non-interacting (blue) patches for the test set (13338 positive pairs and 13338 negative pairs). MaSIF-search was trained and tested on both geometric and chemical features. c. Comparison of the performance between different fingerprint features shown in ROC AUC (13338 positive pairs and 13338 negative pairs from test set). GIF: ROC AUC for GIF fingerprint descriptors [178] Geom: MaSIF-search trained with only geometric features; Chem: MaSIF-search only with chemical features; G+C: geometry and chemistry features. d. (top) Schematic of MaSIF-search workflow showing the 3 stages of the protocol. (bottom) MaSIF-search benchmarking by performing a large-scale docking of N binder proteins to N known targets with site information, results of which are shown in Table 2.1 and Table 2.2.

we obtained ROC AUCs of 0.90 and 0.97, respectively. It is remarkable that chemical features alone can provide such a high discriminative power, the improvement from 0.97 to 0.99 is highly significant, as if we interpret ROC AUC as error probability, it translates to reducing the number of mistakes from 3/100 to 1/100. We next investigated whether inverting the numerical features of the target patch is essential for MaSIF-search. Doing so results in faster learning and in gains in performance in a network trained with all features (ROC AUC of 0.97 with no inversion vs. 0.99 with inversion, Supp Fig. 2.10). Finally, we observed that MaSIF-search and GIF descriptors, have superior performance on high shape complementarity patches, as training/testing on interacting patches with lower shape complementarity results in lower performance (Supp. Fig. 2.10).

Next, we used MaSIF-search to predict the structure of known protein-protein complexes. Ideally, one would be able to predict whether two proteins interact simply by comparing their respective fingerprints, avoiding a time-consuming, systematic exploration of the 3D docking space. We find that fingerprint descriptors can provide an initial and fast evaluation of candidate binding partners. However, a better performance can be achieved by including a subsequent stage where candidate patches (referred to as decoys) selected by the Euclidean fingerprint distance of the patches center points to the target patch are rescored using fingerprints of neighboring points within the patch. Specifically, the MaSIF-search workflow entails two stages (Fig. 5d): I) scanning a large database of descriptors of potential binders and selecting the top decoys by descriptor similarity; II) three-dimensional alignment of the complexes exploiting fingerprint descriptors of multiple points within the patch, coupled to a reranking of the predictions with a separate neural network (see Methods and Supp. Fig. 2.11). The first stage is performed extremely quickly; consequently, MaSIF-search runtime performance is dominated by the second stage, whose complexity depends linearly on the number of decoys used. The tradeoff lies between increasing the number of decoys to improve accuracy, but slow down the overall runtime.

To benchmark MaSIF-search we simulated a scenario where the binding site of a target protein is known, and one attempts to recapitulate the true binder of a protein among many other binders. Specifically, we benchmarked MaSIF-search in 100 bound protein complexes randomly selected from our testing set (disjoint from the training set). For each complex, we first selected the center of the interface in the target protein (see Methods), and then attempted to recover the bound complex within the 100 binder proteins comprising the test set (Fig. 5d). A successful prediction means that a predicted complex with an interface Root Mean Square Deviation (iRMSD) of less than 5 Å relative to the known complex is found in a shortlist of the top 100, top 10, or top 1 results. For comparison, we performed the same task using: PatchDock [48]; ZDock [128, 106]; and ZDock in combination with the scoring application ZRank2 [126] (ZDock+ZRank2). For each program we compared our runtime performance and number of recovered complexes (Table 2.1). Among the baseline tools, PatchDock showed the fastest performance, while ZDock+ZRank2 showed the best performance. MaSIF-search with only 100 decoys per target shows performances similar to PatchDock, but the entire benchmark is performed in just 4 CPU minutes, compared to 2743 CPU minutes for PatchDock. If we

expand MaSIF-search's decoys to 2000, it achieves similar performances to ZDock+ZRank2 with much faster runtimes (4000-fold).

Method	# solved complexes in Top 100	Top 10	Top 1	Time (min)
MaSIF-search Decoys = 100	37	36	30	4
MaSIF-search Decoys = 2000	67	56	43	39
PatchDock	43	32	21	2743
ZDock	58	36	18	134934
ZDock+ZRank2 Decoys = 200000	77	63	45	159902

Table 2.1: Results for large scale docking benchmark benchmark for PatchDock, MaSIF-search (with multiple numbers of decoys), ZDock, and ZDock+Zrank2 on bound (holo) complexes.

Method	# solved complexes in Top 1000	Top 100	Top 10	Time (min)
MaSIF-search Decoys = 2000	17	7	2	16
PatchDock	11	4	1	560
ZDOCK	17	13	5	13174
ZDock+ZRank2 Decoys = 80000	23	12	5	16866

Table 2.2: Results for large scale docking benchmark benchmark for PatchDock, MaSIF-search (with multiple numbers of decoys), ZDock, and ZDock+Zrank2 on unbound (apo) complexes.

Even though we trained only on co-crystallized protein complexes, we also tested our method in a benchmark set of 40 proteins crystallized in the unbound (apo) state. Since unbound docking is significantly more challenging, we changed the success criteria to finding the correct complex within the top-1000, top-100, and top-10, for all methods (Table 2.1). Here the performance of all tools deteriorates, with slightly better accuracy for ZDock and ZDock+ZRank2. Although MaSIF-search can recover many of the complexes within the top 1000 results, the scoring neural network, which was trained on holo structures, does not rank these into the top 10. These results point to the need of training MaSIF on apo structures, perhaps by augmenting datasets with simulated unbound states.

In the previous docking comparison, we provided the site of the interface as input; however, when the target site is unknown, a combination of MaSIF-site and MaSIF-search to predict protein complexes is an attractive possibility. To provide a specific example, we selected the protein complex PD1:PD-L146 (PDB id: 4ZQK) as a test case. We first used MaSIF-site for binding site prediction in the uncomplexed PD-L1 from the co-crystal structure, followed by MaSIF-search to scan a database of 11,000 query structures (52 million surface fingerprint descriptors) in order to find putative binders of the predicted binding site in PD-L1 (this protocol is shown in Supp. Fig. 2.12). The ground truth binder, PD1 was included amongst

the 11,000 structures and PD1:PDL1 related complexes were excluded from the training set. Our combined approach identified the mouse version of PD1 bound to human PD-L1 as the best binder (ranked #1, #3, #4), and the ground truth human PD1 binder (ranked #8) in 26 minutes. Performing vast searches using traditional docking tools is prohibitively expensive. In summary, MaSIF-search identifies patterns that drive protein-protein interactions which are embedded in a space amenable for fast searches.

2.5 Discussion

The molecular surface representation describes the features of a protein that contact other biomolecules, while abstracting the underlying protein sequence. This abstraction allows MaSIF to learn patterns that are independent of a protein's evolutionary history. Crucially, our general approach to learning surface fingerprints may enable a more complete understanding of protein function. This may prove critical in fields of protein science that have been shifting away from naturally evolved proteins. We foresee that MaSIF will be especially important for de novo protein design [75] applications, where the design of new biomolecular interactions remains a fundamentally unsolved problem, despite notable advances [89]. In the future, protein design programs such as Osprey [72] and Rosetta [103] may become fingerprint-aware, optimizing the sequence of de novo-designed proteins to display molecular surface patterns necessary to perform a functional task.

The proof-of-concept applications presented here meant to showcase MaSIF's generality and the concept of learning from surface features. Despite their early-stage development, these methods can be useful to the wide community focused on understanding structure-function relationships. Such applications may entail the characterization of large-scale ligand-protein interaction networks (MaSIF-ligand), identification of "surface hot-spots" which may be more easily targeted for the design of novel biologics for therapeutic purposes (MaSIF-site). MaSIF-search could be coupled to experimental methods to identify binding partners for proteins, or it could be used to find potential engaging partners to use as starting points for protein design [57]. Moreover, all these methods could benefit from sequence evolutionary data to improve their predictive capabilities.

Collectively, we present a conceptual framework to decipher interaction fingerprints, leveraging the representation of protein structures as molecular surfaces, together with powerful data-driven learning techniques. The availability of our data and code will allow researchers to apply our framework to new problems. Our current applications show important technical advantages with great potential for further development and considerable impact on the fundamental study of protein structure and function, as well as for the design of novel proteins and protein-based therapies.

2.6 Methods

2.6.1 Computation of molecular surfaces

All proteins in the datasets were protonated using Reduce [171], and triangulated using the MSMS program [143] with a density of 3.0 and a water probe radius of 1.5 Å. Protein meshes were then downsampled and regularized to a resolution of 1.0 using pymesh51. Geometric and chemical features were computed directly on the protein mesh, with the exception of the distance-dependent curvature, which was computed on each patch according to the surface normals of the vertices in the patch.

2.6.2 Decomposition of proteins into overlapping radial patches and computation of features

For each point in the discretized protein surface mesh, a radial patch of geodesic radius 9 Å or 12 Å (application-dependent) was extracted to perform an analysis of the surface features of the patch. The choice of radius was empirical, mainly driven by performance and memory constraints. For MaSIF-search we chose 12 Å because we found this to be a good value to cover the buried surface area of many PPIs. This patch size was reused for MaSIF-ligand. A patch of 9 Å was selected for MaSIF-site because the smaller patch allowed us to do multiple convolutional layers within our available memory resources, which we found critical for this application. In the absence of memory constraints, a patch larger than 12 Å would be ideal, as MaSIF's geometric deep learning architecture is capable of assigning different weights to different geodesically-clustered kernels.

The following features were included in each patch:

Shape index - describes the shape around each point on the surface, with respect to the local curvature [178]. Values range from -1 (highly concave) to +1 (highly convex). It is defined with respect to the principal curvatures $\kappa_1, \kappa_2, \kappa_1 \geq \kappa_2$ as:

$$\frac{2}{\pi}\tanh^{-1}\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2} \tag{2.1}$$

Distance-dependent curvature - for every vertex within an extracted patch, the distance-dependent curvature computes a value in the range [-0.7, 0.7] that describes the relationship between the distance to the center and the surface normals of each point and the center point. Details of this feature are described in [178]. While the principal curvature component describes the shape around each vertex in the full protein, we found that it is also informative to compute the curvature within each patch, using the center of the patch as a reference.

Poisson-Boltzmann continuum electrostatics - PDB2PQR52 was used to prepare protein files for

electrostatic calculations, and APBS53 (version 1.5) was used to compute Poisson-Boltzmann electrostatics for each protein. The corresponding charge at each vertex of the meshed surface was assigned using Multivalue, provided within the APBS53 suite. Charge values above +30 and below -30 were capped at those values and then values were normalized between -1 and 1.

Free electrons and proton donors - the location of free electrons and potential hydrogen bond donors in the molecular surface was computed using a hydrogen bond potential [93] as a reference. Vertices in the molecular surface whose closest atom is a polar hydrogen, a nitrogen, or an oxygen were considered potential donors or acceptors in hydrogen bonds. Then, a value from a Gaussian distribution was assigned to each vertex depending on the orientation between the heavy atoms [93]. These values range from -1 (optimal position for a hydrogen bond acceptor) to +1 (optimal position for a hydrogen bond donor).

Hydropathy - Each vertex was assigned a hydropathy scalar value according to the Kyte & Doolittle [99] scale of the amino acid identity of the atom closest to the vertex. These values, in original scale ranged between -4.5 (hydrophilic) to +4.5 (most hydrophobic) and were then normalized to be between -1 and 1.

2.6.3 Computation of geodesic polar coordinates

Once surface patches are extracted from a protein, MaSIF uses a geodesic polar coordinate system to map the position of vertices in radial (i.e. geodesic distance from the center) and angular coordinates (i.e. angle with respect to a random directions) with respect to the center of the patch (Fig. 1c). These coordinates add information on the spatial relationship between features to the learning method.

Geodesic distances - On a continuous surface, a geodesic is the shortest path (curve) connecting two points when 'walking' over the surface; geodesic distance between two points is the length of a geodesic between them. On a mesh (the discretization of the continuous molecular surface we use in our implementation), a geodesic is the shortest polyline between two vertices, traversing triangular faces. On a graph, geodesic is a collection of adjacent graph edges connecting two vertices. The computation of geodesics on meshes can be computed exactly or approximated using fast-marching methods, for computational efficiency, we used graph geodesics with weighted edges (corresponding to the Euclidean distance between the vertices), computed using the Dijkstra algorithm, as an approximation to the true geodesic. Since the molecular surfaces were regularly meshed, we found this to be an accurate compromise.

Radial coordinates - Describe the geodesic distance of a point to the center of the patch. Due to its speed, we used the Dijkstra algorithm implemented in Matlab to compute an approximation of the true geodesic distance. Thus, in our implementation the geodesic distance is the sum of the edge lengths that connect the nodes defined on the surface mesh graph.

Angular coordinates - A classical multidimensional scaling algorithm54 implemented in Matlab was used to flatten patches into the plane based on the Dijkstra approximation to pairwise geodesic distances between all vertices. As molecular surface patches have no canonical orientation, a random direction in the computed plane was chosen as a reference, and the angle of each vertex to this reference in the plane was set as the angular coordinate.

2.6.4 Geometric deep learning on a learned soft polar grid

Geometric deep learning allows us to apply successful image-based deep neural network architectures, such as convolutional neural networks (CNNs) [97], to geometric data such as surfaces. Traditional CNNs used in image analysis can be thought of as running a sliding window through the image; at each position of the window, a patch of pixels is extracted. Each pixel is then multiplied by a respective learnable filter value, and the results summed up. On protein molecular surfaces, we do not have a regular grid, hence we replace it with a system of Gaussian kernels defined in a local geodesic polar system of coordinates that act as "soft pixels". The parameters of the Gaussians are learnable on their own [97]. Thus, we refer to this system of Gaussian kernels as a learned *soft polar grid*.

Our learned polar grid contains θ angular bins, and ρ polar bins, for a total of $J = \rho \cdot \theta$ bins. For each vertex in the discretized molecular surface x, with neighbors $N(x)$, and each vertex $y \in N(x)$, we define the coordinates $u(x,y)$, the radial and angular coordinates of y with respect to x. The mapping of each grid cell j for feature vector f and the patch centered at x, $D_j(x)f$ is defined as:

$$D_j(x)f = \sum_{y \in N(x)} w_j(\mathbf{u}(x,y))f(y), \qquad j = 1,\ldots,J \tag{2.2}$$

where w_j is a weight function, and $f(y)$ are the features at vertex y.

Rotation invariance - Is handled in the neural network by performing θ rotations of the input patch and performing a max-pool operation on the output [97].

2.6.5 MaSIF-ligand - ligand site prediction and classification

Dataset – Proteins that bind to the selected cofactors were downloaded from the PDB and their biomolecular assemblies were built using SBI [24].. Details on pocket selection and clustering by sequence are presented in Supplementary note 1.

Neural network architecture, cost function and training optimization - The training step and network architecture was as follows: 32 patches were randomly sampled from a single binding pocket. Each patch was used as input in a network and mapped to a learned soft grid with 16

angular bins and 5 radial bins. Each feature type (2 geometric and 3 chemical features) was run through a separate neural network channel, where the learned soft grid layer was followed by a convolutional layer with 80 filters, an angular max pooling layer with 16 rotations, a rectified linear, and a fully connected layer. A fully connected layer then combined the output from each channel, and output to an 80-dimensional fingerprint. The resulting 32 fingerprints were multiplied together to generate an 80x80 covariance matrix. The architecture for this network is shown in Supp. Fig. 2.14. The covariance matrix was flattened and fed first to a 64-unit, fully connected layer with rectified linear activation, and then to a 7-unit, fully connected layer with linear activation, followed by a softmax cross-entropy loss. The network was trained for 20000 iterations (rather than epochs) with the Adam optimizer with a learning rate of 1E-4. The validation error was evaluated every epoch and the best network was selected based upon this value. The initial choice of randomly sampling 32 patches in the pocket was made for three reasons: (i) each patch covers a 12 Å radius, and thus, 32 patches are likely to cover the surface from the entire pocket; (ii) the number is low enough so that all ligand types are in contact with at least these many patch centers, and (iii) due to memory restrictions, since a larger number of patches exceeds our GPU memory capabilities. To obtain more stable predictions, each pocket was sampled 100 times and the resulting 100 predictions were averaged to obtain the final prediction.

Visualization of relevant patches for NADP/NAD discrimination in Supp. Fig. 2.7 - See supplementary note 2.

Comparisons to SiteEngine [152], ProBIS [92], and KRIPO [139] - See supplementary note 3.

2.6.6 MaSIF-site - protein interaction site prediction

Datasets - Protein-protein interaction pairs were taken from the PRISM list of nonredundant proteins, the ZDock benchmark, PDBBind, and SabDab [16, 111, 49, 166]. Sequence splits were performed using CD-HIT60 and structural splits were performed using TM-align61. Details on the sequence and structural split are described in supplementary note 4.

Definition of interface points in a protein surface - We defined the ground truth interface as the region of the surface that becomes inaccessible to solvent molecules upon complex formation. This was done by computing the surfaces of the complexes and the unbound partners. Surface regions in the individual partners that have no corresponding surface in the bound complex were then defined as the ground truth interface. Surface regions that become solvent inaccessible upon complex formation were defined as the ground truth interface.

Neural network, cost function, and training optimization - A neural network with three convolutional layers was used for this application. A diagram of the architecture is shown in Supp. Fig. 2.15. The network received as input a full protein decomposed into overlapping surface patches with a radius of 9.0 Å. The smaller patch radius was selected because it reduced memory requirements, thus allowing more convolutional layers. The patches are mapped

onto learned grids with 3 radial bins and 4 angular bins. The output of the network is an interface score between 0 and 1 for each patch center point. During training, the batch size consisted of a single protein, and the network was optimized using an Adam optimizer [90] on a sigmoid cross-entropy loss function. As the number of non-interface points is usually much larger than the number of interface points, a random subset of non-interface points was selected to train on an equal number of positive and negative samples. Training of the neural network was performed during 40 'wall clock' hours, after which the job was automatically killed. These 40 hours allowed for 43 epochs, whereas in each epoch all proteins in the training set were fed to the network. The best model was saved whenever the validation set's ROC AUC improved over that of a previous model. The last saved model occurred at epoch 42, which indicates that the neural network could have continued learning beyond the 40 allotted hours.

Comparisons to PSIVER [121] and SPPIDER [130] - See supplementary note 5.

2.6.7 MaSIF-search - prediction of PPIs based on surface fingerprints

Datasets - Details on the dataset and split are presented in supplementary note 6.

Selection of interacting and non-interacting patches - For each PPI, all pairs of surface patch centers belonging to distinct proteins and within 1.0 Å of each other were considered further. A radial shape complementarity score was computed for the pair as follows: (i) the shape complementarity of each point in the patch to the neighboring patch was computed; (ii) points within 12 Å of the center were divided into 10 concentric radial bins, in increments of 1.2 Å; the shape complementarity of the bin was computed as the 25th percentile of the points in the bin; (iii) the radial shape complementarity S of the patch was computed as the median across all bins. The neural network for Fig. 5 was computed with interacting patches with a value of $S > 0.5$, while different ranges of S ($-1 < S < 0.1$ for very low complementarity, $0.1 < S < 0.3$ for low complementarity, and $0.3 < S < 1.0$ for high complementarity) were also used to train and test (Supp. Fig. 2.10). Non-interacting pairs were selected by pairing a truly interacting patch with a randomly chosen one from any other protein in the set.

Neural network architecture, cost function and training optimization - The MaSIF-search neural network receives the features of one patch (which may be inverted for the binding partner) as input and then outputs a vectorized descriptor. The architecture for this network is shown in Supp. Fig. 2.16. During training and testing, a binder, a target and a random patch are input into the network, such that the binder and target are known interacting pairs, and the target and random are assumed to be non-interacting. The features for the target are inverted (multiplied by -1), with the exception of the hydropathy index. A total of 85652 true interacting pairs and 85652 non-interacting pairs were chosen for training/validation, while 12678 true interacting and 12678 non-interacting pairs were chosen for testing. The network was trained to minimize the Euclidean distance between the fingerprint descriptors of binder and target, and maximize the distance between the descriptors of target and random. Each patch was input to a network and mapped to a learned soft grid with 16 angular and 5 radial

bins. Each feature type (2 geometric and 3 chemical features) was ran through a separate neural network channel, where the learned soft grid layer was followed by a convolutional layer with 80 filters, an angular max pooling layer with 16 rotations20, and a rectified linear unit. A fully connected layer then combined the output from each channel, and output an 80-dimensional fingerprint. The optimization process during training, using an Adam optimizer [90], consists of minimizing the d-prime cost function63:

$$f(x) = \sigma_t + \sigma_f + \mu_t + \max(0, M - \mu_f) \tag{2.3}$$

where μ_t and μ_f are the median distance for true and non-interacting pairs, respectively, while σ_t and σ_f are the standard deviation for true and false interacting pairs. The neural network was trained with batches consisting of 8 binder, 8 target, and 8 random patches. In each batch the true interacting pairs and the random patch were randomly selected. The network was trained for 40 'wall-clock' hours, and killed after 40 hours, which allowed for 335000 iterations. The validation sets were evaluated after every 1000 iterations. The best neural network model was determined as the one where the ROC AUC on the validation set achieved a maximum, which was reached after 260000 iterations.

Structural alignment and rescoring - A second-stage alignment and scoring method generates the complexes based on the identified fingerprints. The top decoy patches with the shortest fingerprint descriptor distance to the target patch are selected as a short list of potential binding partners. Each binder patch is then aligned using the RANSAC algorithm implemented in Open3D64 (Supp. Fig. 2.11). Briefly, RANSAC selects three random points from the binder patch and uses the computed descriptors to find the closest points in the target patch by descriptor distance. Using these three newly found correspondences, RANSAC attempts to align the source patch to the target patch. RANSAC iterates 2000 times and selects the transformation with the highest number of points within 1.0 Å between binder and target. Following RANSAC, an additional algorithm, the iterative closest point algorithm, as implemented in Open3D optimizes the alignment. After RANSAC completes, the transformation is rescored with a separate neural network. To optimize speed, the extracted patches were reduced to 9 Å.

Neural network for scoring aligned patches – To discriminate true alignments we trained a separate neural network to score binder patches after the alignment step (Supp. Fig. 2.11). Once a patch alignment has been made, the nearest neighbor on the binder in 3D space to each point in the target is searched, establishing correspondences (Supp. Fig. 2.11b). Then, the input to the neural network is the 3D Euclidean distance, the MaSIF-search fingerprint distance and the product of the normals between correspondences. The output is a predicted score on the alignments. To train this neural network we generated thousands of true and false alignments in the MaSIF-search training set. For each target structure we used one true alignment (defined as the true binder aligned within 5 Å iRMSD accuracy) and 200 false alignments (either sourced from a different protein from the true binder, or from the same protein but with over 5 Å iRMSD). iRMSD was defined as the RMSD of the C_α atoms of the binder that were less than 10 Å away from any of the C_α atoms of the target. For each point

in an aligned patch we found its nearest neighbor (in 3D space, after alignment) on the target patch; for each pair of (binder, target) points we measured: MaSIF-search fingerprint descriptor distance; the Euclidean distance in 3D space; dot products between their normals. The input features to our network were: 1/(descriptor distance), 1/(Euclidean distance), and the dot product of the normals. Each aligned patch was limited to 200 points, if the size of the aligned patch was greater than 200 points it was randomly sampled, and if it was less than 200 points it was zero-padded. Thus, the input to the network is a matrix of size 200,3 (200 point pairs with three features per pair). The network architecture was as follows: series of 1-dimensional convolutional layers of dimensionalities 8, 16, 32, 64, 128, 256 with all these layers having a kernel size and stride of 1; this was followed by a global average pooling layer and then a series of fully connected layers of dimensionality 128, 64, 32, 16, 8, 4, 2; alignments were labeled as positives or negatives and a cross-entropy loss was used, the negative class was weighted with 1/200. The Adam optimizer was used with a learning rate of 1e-4. From the training set, 10% of alignments were used as a validation set, the network was trained for 50 epochs with a batch size of 32. The best model was selected based on the lowest validation loss.

PPI search docking benchmark - See supplementary note 7.

Comparisons to GIF Descriptors [178], PatchDock [48], Zdock [128], and ZRank2 [127] - See supplementary note 8.

PDL1 benchmark - See supplementary note 9.

2.6.8 Pre-computation and neural network running times

The precomputing time of the PDB files to generate surfaces with features and runtime for MaSIF-search and MaSIF-site neural networks is dependent on the protein size, and is thus plotted in Supp. Fig. 2.17. For example, a 125 amino acid protein is processed in 99.4 s accounting CPU, System and GPU times. GPU times were measured using 'wall-clock' time, since standard UNIX time tools do not account for GPU processing time. All times were measured on an Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz, and an NVIDIA Tesla K40 GPU running Red Hat Enterprise Linux 7.4. PDB files precomputations were performed on CPUs, while neural network calculations were performed on GPUs.

2.6.9 Data availability

Datasets - The bound PDBs in the training/testing set and the computed surfaces with chemical features are available at Zenodo with DOI: 10.5281/zenodo.2625420. The unbound PDBs in the test set are provided in the github repository. All scripts to generate the datasets are available at https://github.com/lpdi-epfl/masif.

Code availability - All code was implemented in Python and Matlab. Neural networks were

implemented using TensorFlow [1]. Both the code and scripts to reproduce the experiments of this paper are available at: https://github.com/lpdi-epfl/masif67. The github repository also provides a PyMOL68 plugin for the visualization of feature-rich molecular surfaces, used for the Figures in this paper. All source code is provided under an Apache 2.0 permissive free software license.

2.7 Supplementary

2.7.1 Supplementary figures

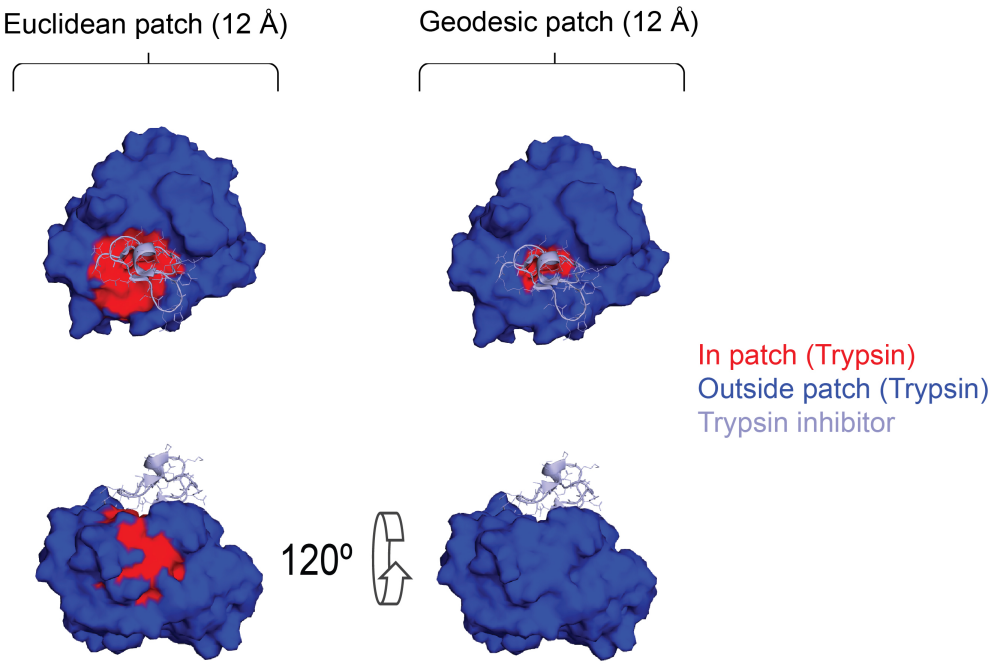


Figure 2.6: Example-based illustration on the importance of geodesic distances in modeling protein surfaces. This example shows Trypsin (blue/red surface) in complex with the (cyan cartoon+line representation) (PDB ID 1PPE). We selected a point in the deep pocket of the interface, and colored in red every surface point within a 12 Å Euclidean radius-defined patch (left) or a 12 Å Geodesic radius-defined patch (right). The Euclidean patch (left, below) includes points on a different face of the protein, far from the binding site, while the geodesic patch only includes points in the face that interacts with the protein. This example shows that, especially in highly irregular surfaces the geodesic distances between points can be much larger than the Euclidean distances and that in such cases geodesic distances can be more relevant.

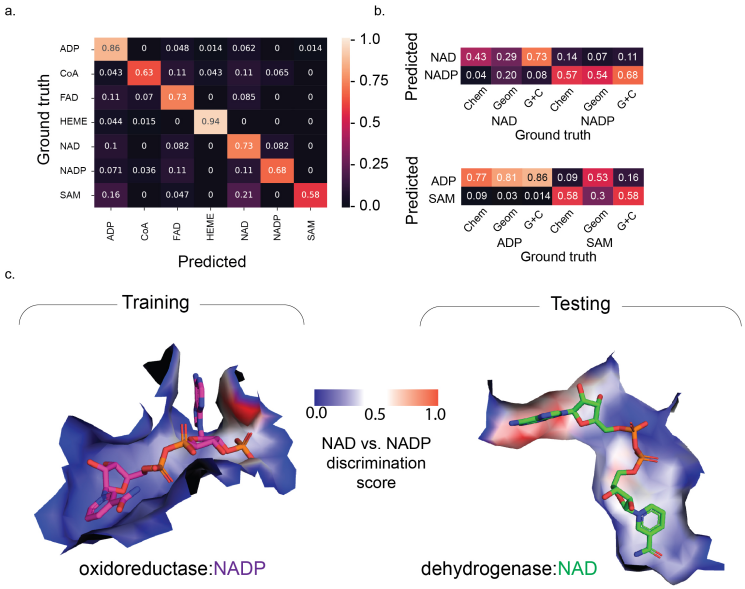


Figure 2.7: Analysis of MaSIF-ligand performance for specific cofactors. a. Confusion matrix of ligand specificity on a MaSIF-ligand neural network trained with all features. Number of pockets in each category: ADP:146, CoA:46, FAD:71, HEME:68, NAD:49, NADP:28, SAM:43. b. Subset of the confusion matrices showing the importance of the features in distinguishing pockets between highly similar ligands. Number of pockets in each category: ADP:146, NAD:49, NADP:28, SAM:43. c. Analysis of MaSIF-ligand's discrimination between NADP and NAD on two specific examples: a bacterial oxidoreductase and a human dehydrogenase. The bacterial dehydrogenase in the test set binds to NAD (PDB ID 2O4C), while its closest structural homologue in the training set corresponds to a mammalian oxidoreductase (PDB ID 2YJZ), which binds to NADP. Here we scored the pocket surface by a discrimination score, which scores each point in the protein surface by its weight in the neural network's distinction between NADP and NAD. Surface regions with high importance are shown in red, while those of low importance are shown in blue.

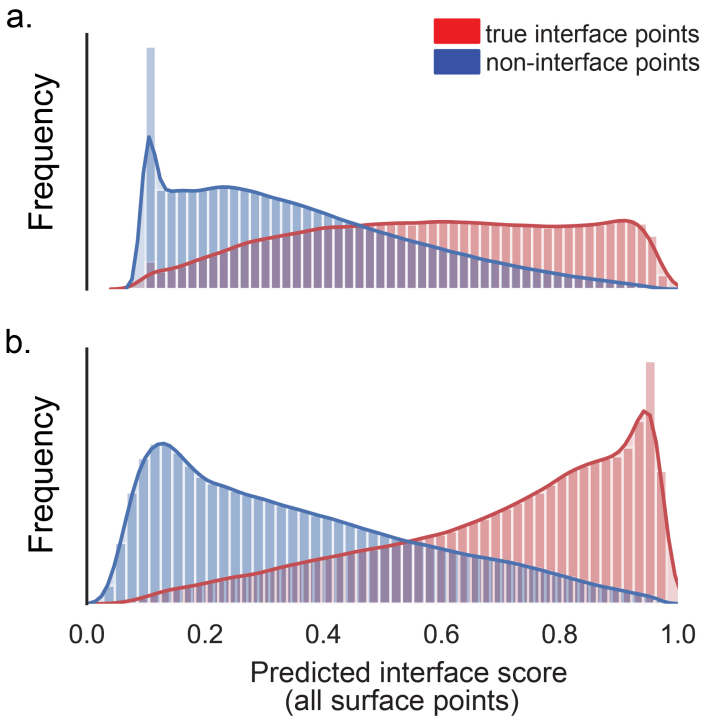


Figure 2.8: MaSIF-site interface prediction score distribution for true positives (red) vs. true negatives (blue). a. One convolutional layer obtains a ROC AUC value of 0.77 (n = 2192870 points from the test set) and b. Three convolutional layers obtain a ROC AUC value of 0.86 (n = 2192870 points from the test set).

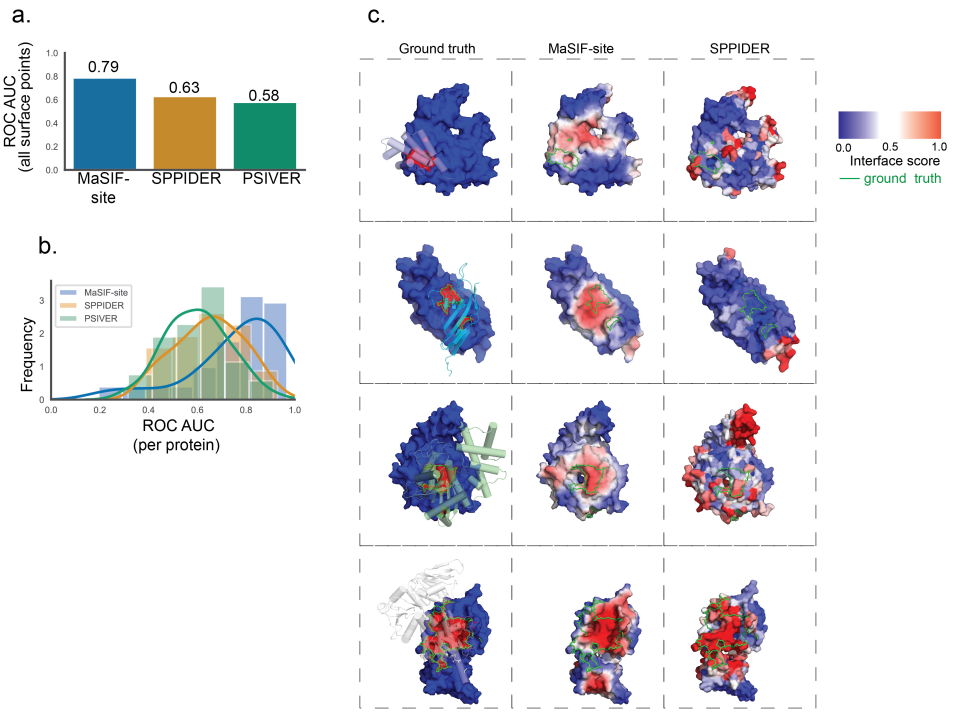


Figure 2.9: Comparison between MaSIF-site and two other predictors on a set of transient interactions. a. ROC AUC values over all surface points of MaSIF-site vs. SPPIDER vs. PSIVER on 53 proteins involved in transient interactions. b. Histogram showing the distribution of ROC AUCs per protein for the 53 proteins on a residue basis for MaSIF-site, SPPIDER and PSIVER. c. Randomly-selected examples from the testing set comparing MaSIF-site prediction with SPPIDER.

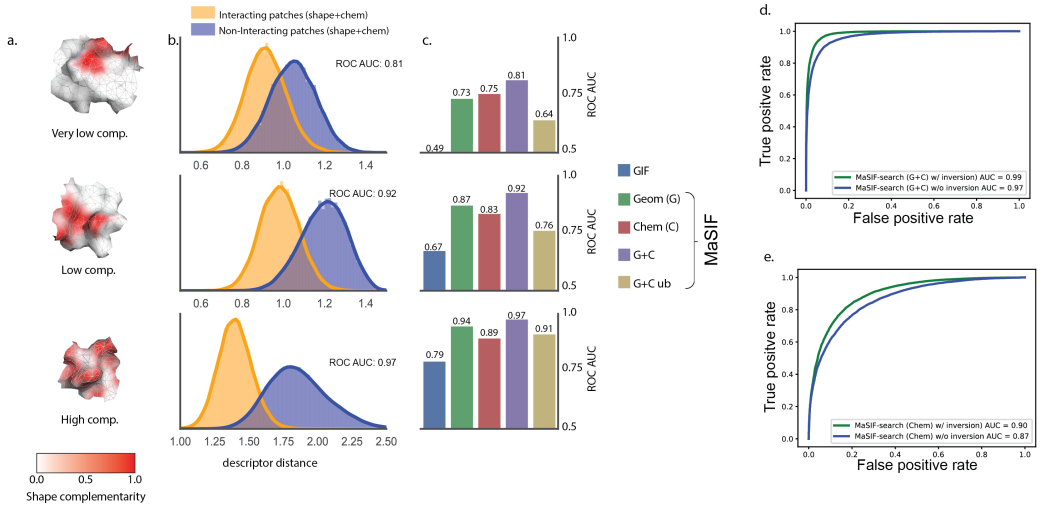


Figure 2.10: Performance of MaSIF-search fingerprints under different shape complementarity filters for the interacting patches, and effect of inverting input features. a. We set up three classes of interacting patches, filtered by shape complementarity, and trained neural networks with each set. The sets are illustrated here with three examples, where the surface is colored according to shape complementarity from white (0.0) to red (1.0). b. Descriptor distance distribution plot for interacting and non-interacting patches depending on the shape complementarity class. c. ROC AUC values for the GIF descriptors, MaSIF descriptors trained only on geometry, chemistry, or both, and patches found in unbound proteins within each complementarity class (G+C ub). # of pairs of patches: high comp, 38038 positives and 38038 negatives; low comp.: 16798 positives and 16798 negatives; low comp. 21297 positive and 21297 negatives. d-e. MaSIF-search benefits from the inversion of features in the input. d. ROC AUCs of a network trained/tested with inversion (green) vs. a network trained/tested without inversion (blue) using both Geometric (G) and chemical (C) features. The plot's ROC curve was computed on 13338 positive and 13338 negative pairs of samples. e. Performance of a network where electrostatics and the hbond features were inverted (green) vs. one in which they were not (blue), on a network trained with only chemical features.

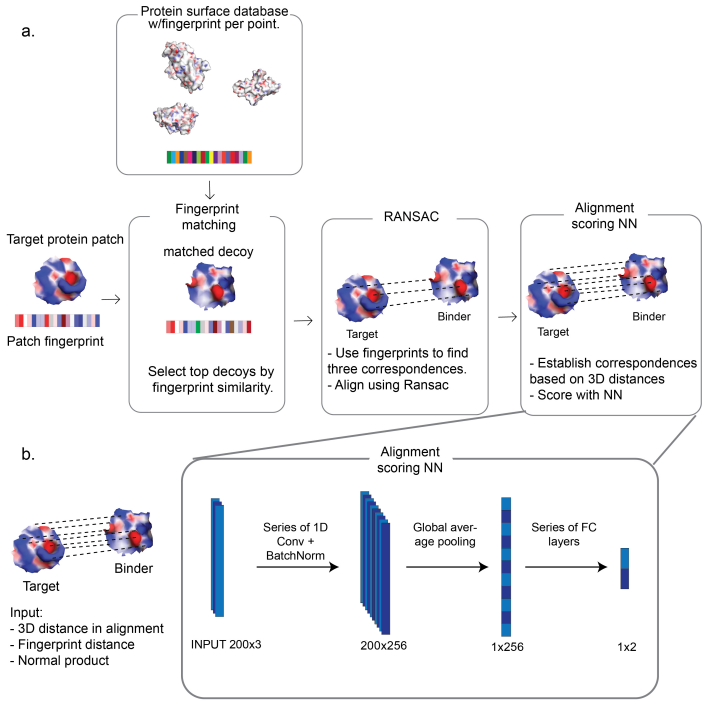


Figure 2.11: MaSIF-search protocol for the generation of protein complexes. a. A fingerprint is computed on a selected target site (left). A database of proteins with precomputed fingerprints is searched for the K- most similar fingerprints. Once these are matched, a set of correspondences between the matched patches is found with the RANSAC algorithm, which uses the fingerprints of other points in the patch to obtain a good alignment. RANSAC selects the alignment with the most points within 1.5 Å of each other. The transformation is then scored using: Euclidean distances; fingerprint distances; and the normal products between neighboring points (see Methods). b. Neural network architecture for the alignment scoring function. Correspondences are first assigned between the aligned binder and target patches based on the nearest point in 3D space. For every correspondence, the 3D distance between the points, the Euclidean distance between the fingerprint descriptors and the product of their normals is input into the neural network. The input is a matrix of size 200 by 3: the maximum number of points allowed in the patch times the three features. The output is a 2-dimensional logit with the predicted score.

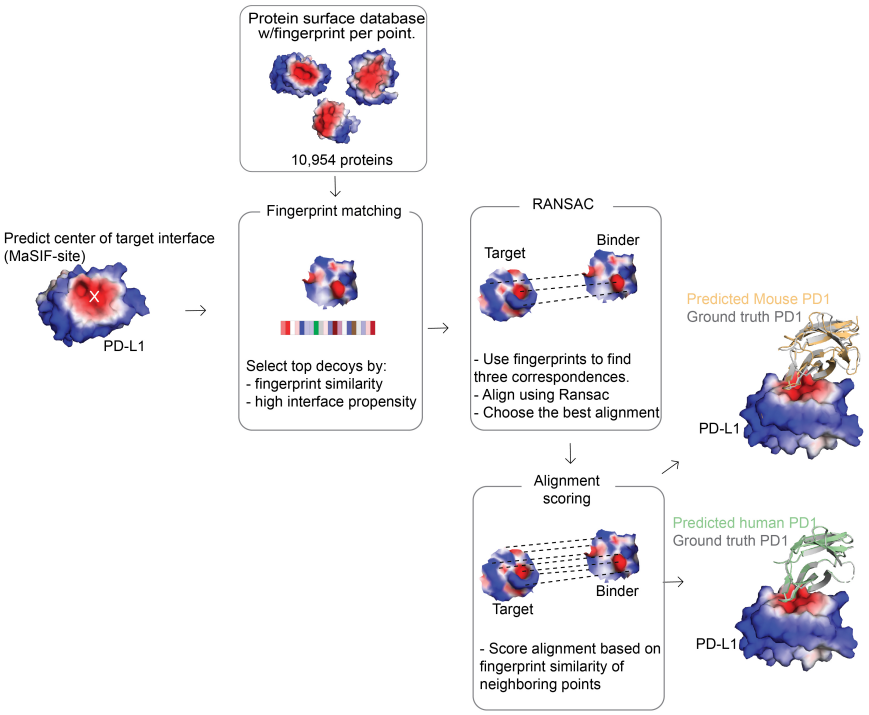


Figure 2.12: Hybrid MaSIF-search/MaSIF-site protocol to identify true binders against PD-L1. The target site is first predicted using MaSIF-site. Then a database of nearly 11,000 proteins is scanned, all patches with a MaSIF-site score > 0.9 and with a descriptor distance less than 1.7 are selected for alignments. Top candidates are matched using RANSAC, and reranked using the descriptor distance of all aligned points (described in Methods). The top predicted complex was the PD-L1:Mouse PD1 (PDB ID 3BIK), ranked #1 with an RMSD of 0.6 Å (shown here in pale orange). The PD-L1:Human PD1 (PDB ID 4ZQK), was ranked #8 with an RMSD of 0.3 Å. Both are shown overlaid over the initial complex (PDB ID 4ZQK). The entire runtime protocol took approximately 26 minutes (excluding descriptor precomputation time).

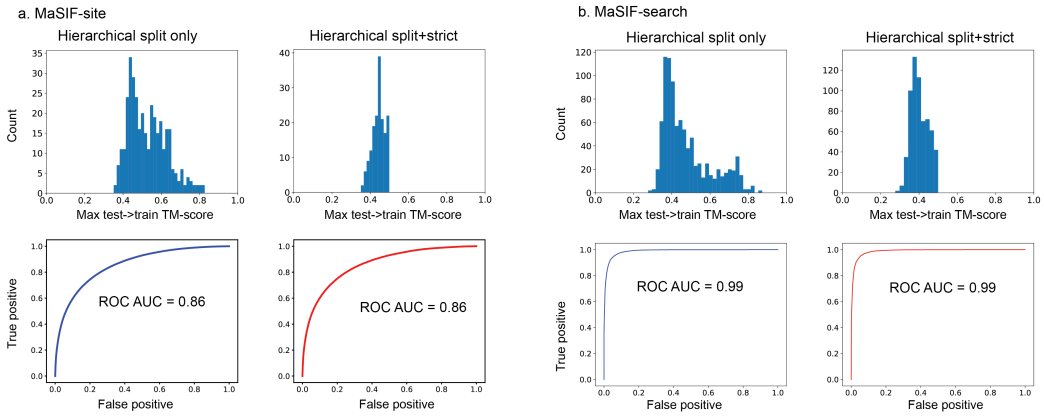


Figure 2.13: The performance of MaSIF-search and MaSIF-site is not affected by a stricter structural split. MaSIF-site and MaSIF-search's test sets were split from the training sets using a hierarchical clustering approach based on a matrix of TM-scores. In the case of MaSIF-search this split was performed using the interface TM-score. (hierarchical split only, a, b, top left). Some structures in the test set still maintain a TM-score above 0.5 to at least one member in the training set. (a,b, top right) We performed a stricter split by eliminating all members of the test set whose maximum TM-score to any member of the training set was above 0.5. (a,b, bottom right). The stricter split did not affect performance. a. MaSIF-site (left) Hierarchical split only test set consists of 359 proteins decomposed into 2191879 patches. (right) Hierarchical split+strict test set consists of 169 proteins decomposed into 1042951 patches. b. MaSIF-search (left) Hierarchical split only test set consists of a total of 957 proteins decomposed into 13338 interacting patch pairs and same number of non-interacting pairs. (right) Hierarchical split+strict consists of 635 proteins decomposed into 7135 interacting patch pairs and same number of non-interacting pairs.

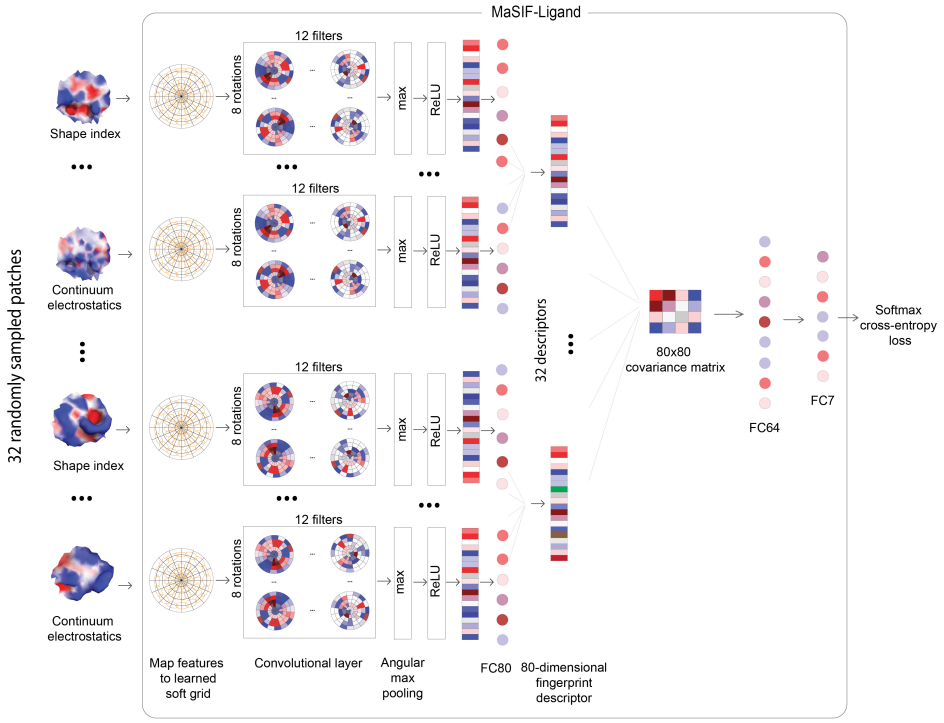


Figure 2.14: Network architecture for MaSIF-ligand. 32 randomly sampled pocket patches are fed through convolutional layers followed by a fully connected layer (FC80). Descriptors are combined in a 80x80 covariance matrix followed by two fully connected layers (FC64 and FC7) and then softmax cross-entropy loss.

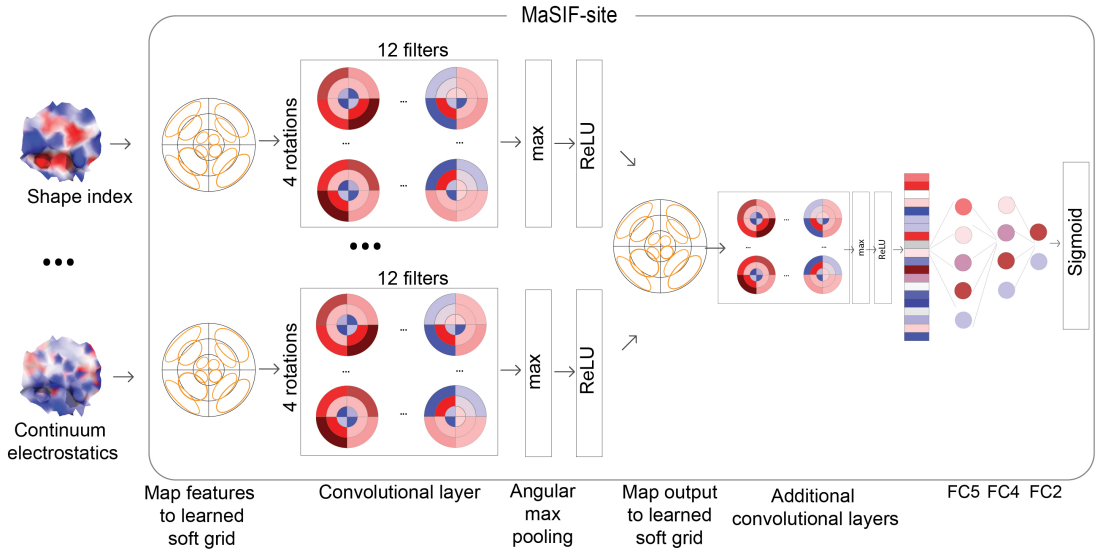


Figure 2.15: Network architecture for MaSIF-site. Patches are fed through convolutional layers followed by a series of fully connected layers (FC5, FC4, FC2), and finally a sigmoid cross-entropy loss.

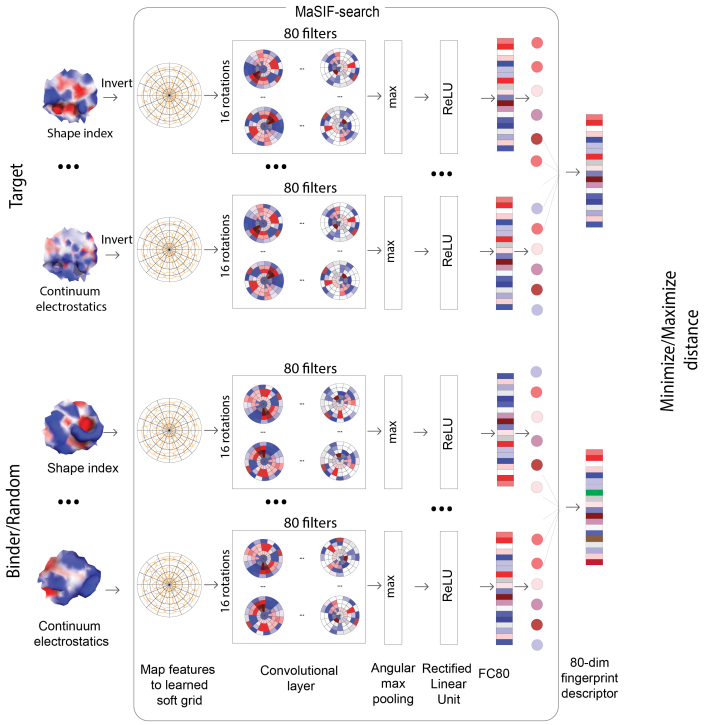


Figure 2.16: Network architecture for MaSIF-search. Patches from the target and the corresponding binder or a random patch are fed through convolutional layers, followed by a fully connected layer (FC80). The L2-distance between the resulting descriptors is computed and the neural network is optimized to minimize this distance with respect to binder and maximize it with respect to the random patch.

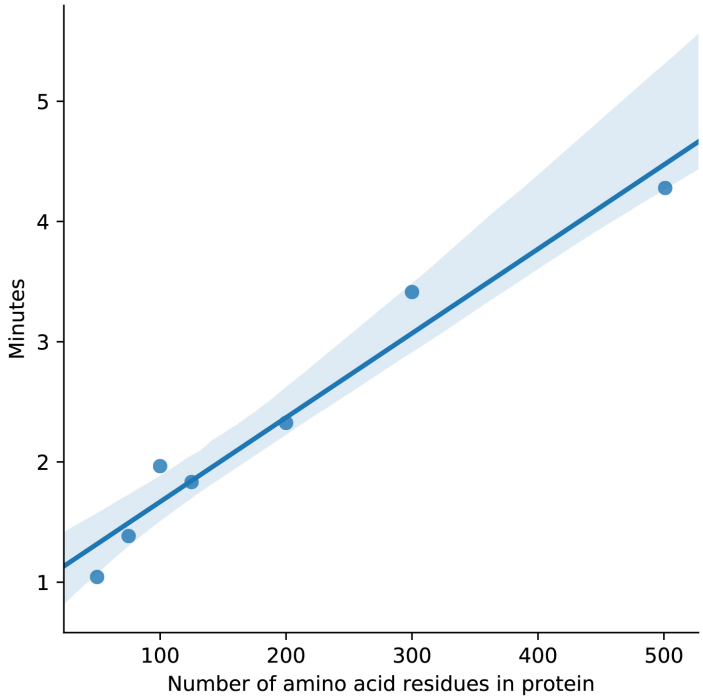


Figure 2.17: Total computation time for MaSIF-search and MaSIF-site for proteins of various sizes. Proteins chains, of sizes: 50, 75, 100, 125, 200, 300, 500, were selected from the PDB. Each chain was run through both the MaSIF- site and MaSIF-search protocols, entailing: downloading the PDB, computing surfaces, input features, and coordinates, decomposing into patches, and computing MaSIF-site predictions and MaSIF-search descriptors. The y-axis shows the CPU user + System time + GPU time in minutes. GPU time consists of the time where the data is processed by the neural network, and was measured in real clock time (i.e. not GPU processor time). The total GPU time is low compared to the overall time, from 4 seconds for a 50-residue protein, to 12 seconds for a 500-residue protein. The line represents the regression fit to the n=7 data points and the shaded area represents the 95% confidence interval.

2.7.2 Supplementary notes

Supplementary note 1. All structures in the Protein Data Bank (PDB; 16 Oct 2018) including a protein chain but no DNA or RNA were considered if they included any of these seven chemical identifiers: ADP, COA, FAD, HEM, NAD, NAP (for NADP) or SAM. This resulted in 1853 ADP structures, 490 COA, 2020 FAD, 4448 HEM, 1269 NAD, 1212 NADP and 393 SAM. After building the biological assembly of these structures, the dataset was filtered based on sequence identity, to reduce redundancy and similarity between structures in the training and test sets. The filtering was performed as follows: the PDB provides pre-computed sequence clusters based on 30% sequence identity; each protein structure in the dataset was associated with one or more of these clusters based on its protein chains; two protein structures were defined as homologous if the associated clusters of both proteins coincided. The dataset was then reduced by randomly sampling structures from the dataset, one at a time, while continuously eliminating their homologs from the sampling pool. This process resulted in a total of 1459 structures, which were then randomly assigned to training (72%), validation (8%) and testing (20%) sets. The surfaces for these structures were generated as described above, and patches of 12 Å radius extracted. If the center point of a patch was less than 3 Å from an atom for any of the seven ligands, the patch was labeled as a part of the binding pocket of the corresponding ligand.

Supplementary note 2. From the NAD binding pocket of the dehydrogenase:NAD pocket (PDB id: 2O4C), 32 patch center points were randomly sampled 10000 times and binding predictions made for each, giving 10000 predictions (7-dimensional vectors). For each prediction the probability ratio NAD/NADP was computed. The predictions giving the top 90th percentile for this ratio were picked and the frequency of the patch centers behind these predictions were computed. The frequencies were normalized and overlaid on the protein surface. Same procedure was applied for the dehydrogenase:NADP complex (PDB id: 2YJZ) except that the NADP/NAD ratio was computed.

Supplementary note 3. *Comparison to KRIPO* – KRIPO was used to generate fingerprints for ligand interactions in all structures from the training and testing set without fragmenting the ligands. Each fingerprint from the testing set was then compared to every fingerprint from the training set. KRIPO does not support the generation of fingerprints for HEME and thus this ligand was removed from the benchmark. Each fingerprint in the testing set was matched against ligand-labeled fingerprints in the training set (ADP, COA, FAD, NAD, NADP and SAM) resulting in six similarity scores for each query fingerprint. These scores were normalized to sum to one, giving a prediction of the ligand-binding preference.

Comparison to ProBiS – The ProBiS program was used to compute scores (z-scores) between each pocket in the test set to all pockets in the training set. For each test set pocket a score was assigned to each ligand type. The score for ligand X (X = ADP, COA, FAD, NAD, NADP and

SAM) is the highest z-score found between the test set pocket and any pocket binding ligand X. We normalized the scores on a per-pocket basis as we found this improved ProBiS's ROC AUC value. The program was run with the -noprune flag to score all pockets, and the minimum z-score was set at -1000. To perform a pocket-level structural split (for the results shown in Fig 2e), all residues with an atom within 3.0 Å of a ligand atom were selected as the pocket residue. Then, TM-align was used to align each pocket of the test set to each pocket of the training set. Pockets aligning at TM-score > 0.5 to any element of the training set were eliminated from the structural split. The testing set consisted of all pockets that successfully ran on all three programs.

Supplementary note 4. The PRISM database56 of PPIs, a compendium of non-redundant PPIs found in crystal structures, was used as the first source. Proteins with parsing problems or that failed to complete the feature computation were discarded. The PRISM database contains many complexes formed by the contacting protein chains found in asymmetric crystal units, which likely do not form in solution. To remove those complexes, we discarded PPIs that have no pairs of patches below a minimum threshold of radial shape complementarity (set at S=0.4; see below for a definition). In total, 8466 proteins engaged in PPIs were taken from the PRISM database. In addition, 3536 non-obligate (transient) interactions were taken from three databases: the PDBBind57, the SAbDab antibody:antigen database58 and the ZDock benchmark set59. Finally, the resulting 12002 proteins were clustered according to sequence identity using the psi-cd-hit60, at 30% sequence identity and one representative member was chosen from each cluster, resulting in 3362 proteins. A pairwise matrix of all TM scores for these proteins was then computed, and a hierarchical clustering procedure using scikit-learn (AgglomerativeClustering)61 was used to split the sets, resulting in a training set of 3004 proteins and a testing set of 358 proteins. Using this hierarchical split approach still resulted in some members of the testing set having at least one member in the training set with a TM-score above 0.5. A TM-score above 0.5 means that the proteins assume roughly the same fold. However, upon performing a stricter split by eliminating all members of the testing set that align at TM-score > 0.5 to any member of the training set, we see no difference in performance (Supp. Fig 8).

Supplementary note 5. *Comparison to SPPIDER* – The performance over a set of 53 single chains (from co-crystal structures) involved in known transient interactions for the test set was compared with that of the interface predictor SPPIDER30. Each protein was uploaded to the SPPIDER web site (http://sppider.cchmc.org/) and a regression-based prediction was computed on each residue. Following SPPIDER's definition of ground truth interface residues30 as closely as possible, the ground-truth interface residues were defined as those whose solvent excluded surface changes at least 5 Å2 upon binding and at least 4% change in interface area. We note that we used the solvent-excluded surface for these calculations and not the solvent accessible surface. In order to perform a comparison with MaSIF-site, MaSIF-site's predictions were converted to a per-residue scoring by assigning the maximum score of all

the residue's points in the surface. A ROC AUC comparison on a surface point basis is shown in Supp. Fig. 4a. *Comparison to PSIVER* - The sequence of each of the 53 proteins of the test set was uploaded to the PSIVER server (https://mizuguchilab.org/PSIVER/). The results of PSIVER assign a regression-based score on each amino acid residue of the protein, which was compared with the ground-truth. For both SPPIDER's and PSIVER's predictions in Figure 4, each of the designed proteins was assigned the predicted score as a b-factor in 1-99% scale and colored in PyMOL from a blue to red spectrum.

Supplementary note 6. A dataset of protein pairs that were co-crystallized and shown to engage in PPIs were taken from the PRISM database (see above). In addition, 3536 non-obligate (transient) PPIs were taken as was done for the interface site prediction, forming a set of 6001 PPIs. For MaSIF- search we did not perform a sequence split, since we consider valid that two proteins with very high sequence identity (for example, two antibodies) binding to two completely different targets, can be in the training and testing set without the risk of overfitting. Instead, we perform our split using structural alignments of the interface atoms of each PPI. The PPI structural interface was extracted from the native complexes and a pairwise TM- align63 score matrix with all interfaces was computed. Then, a hierarchical clustering of the structures was performed according to the TM-align score using scikit-learn's hierarchical clustering (AgglomerativeClustering)61. In total, the dataset was split into 4944 training PPI pairs and 957 testing PPIs. This list is complemented by 40 apo complexes, corresponding to those proteins in the testing PPIs such that both partners' apo crystal structure was available in the ZDock benchmark, belonging to the 'rigid docking' category59. The list of PDBs in the training and testing sets are provided in our github repository. Using this hierarchical split approach still resulted in some members of the testing set having at least one member in the testing set aligning with a TM-score above 0.5 to some member of the training set. However, upon performing a stricter split by eliminating all members of the testing set that align at TM-score>0.5 to any member for the training set, we see no difference in performance (Supp. Fig 8).

Supplementary note 7. N=100 co-crystal structure complexes were randomly selected from the testing set. One of the two proteins was selected as the target protein; for each target protein, the patch with the highest radial shape complementarity to the binder protein patch in the co-crystal structure was selected as the target site (Fig. 5d). Each binder protein was docked onto each target site. The benchmark consisted of recovering the conformation of the true binder within a short list of the top-ranked results (top-100, top-10, top-1, shown in Fig. 5e). A second benchmark was performed with N=40 complexes in the apo state, aligned to the known bound complex. The benchmark for apo structures was performed in the same way as for the co-crystal structures, but the success criteria were relaxed to recover the conformation of the binder within a larger number of top results (top-1000, top-100, top- 10). For all methods benchmarked, all binders were randomly rotated before performing any alignments.

Supplementary note 8. *Comparison to GIF descriptors* - Geometric invariant fingerprint (GIF) descriptors were implemented to our best efforts according to the description by Yin et al15. For testing of the descriptors, the features of the target were inverted before computing the GIF descriptor.

Comparison to PatchDock - PatchDock40 was used with default settings, assigning the residue closest to the target site as an active site residue. After all alignments, PatchDock's transformations for all targets were merged and ranked according to PatchDock's default Geometric Score. The top solutions (100, 10 and 1) for the bound complexes and unbound complexes (1000, 100, 10) were evaluated for agreement to the ground truth complex. PatchDock's time was measured as CPU usage time.

Comparison to ZDock - ZDock 3.0.241 was downloaded as compiled binaries for Linux 64- bits. The surfaces of each target and binder protein were first marked using the marksur program provided in the ZDock package. ZDock allows the definition of a target site, by allowing the user to 'block' every atom that is not in the target site. Thus, we determined the target site by drawing a 12 Å geodesic patch on the protein surface from the center of the interface. Then, all the atoms directly in contact with the vertices in the patch were added to a set of 'non-blocked' atoms. Every other atom in the protein was then blocked by setting the field in columns 55-56 of the target's pdb file to the code '19' as described in ZDock's user manual. For the bound (holo) benchmark, this process was run 10,000 times (100 binders for each of the 100 targets), while for the unbound (apo) benchmark the process was run 1600 times (40 binders for each of the 40 targets).
For each target:binder pair, ZDock generates, by default, 2000 docking results. Thus, the output files for all binders were merged and resorted by ZDock's score. Then, the top solutions for the bound complexes (100 ,10 and 1) and unbound complexes (1000, 100 and10) were evaluated for agreement to the ground truth complex.
Due to the large computational expense of these many runs, ZDock was run on a Google Cloud server with 96 virtual CPU processors and 360 GB of memory. The task was parallelized by running each target against all binders in its own thread. The time measured was CPU user time over all 10,000 runs for the holo benchmark, and over 1600 runs for the apo benchmark. Although the use of a different processor type could affect the running time comparisons with the PatchDock and MaSIF-search methods, the orders of magnitude difference between the methods is unlikely to vary significantly.
The output docking poses of ZDock41 were used as input to ZRank243. Although the running time of ZRank2 could be reduced by limiting the list of poses from ZDock, we used the entire list as the running time was still dominated by ZDock. The docked poses of the binders and targets were protonated with Reduce51. After ZRank2 was run, all the output results were merged and reranked according to the ZRank2 energy function. The time reported by ZDock+ZRank2 was the total CPU user time of ZDock + the total CPU user time for ZRank2. ZRank2 was also run on a Google Cloud server with 96 virtual CPU processors and 360 GB of memory. The task was parallelized by running each target against all binders in its own thread. The time measured was CPU user time over all 10000 runs for the holo benchmark, and over

1600 runs for the apo benchmark.

Supplementary note 9. The task consisted on recovering the bound PD-L1:PD1 complex, among all possible complexes between PD-L1 and 10954 other proteins. First, the binding site scores on the surface of the PD-L1 chain (chain A in PDB id: 4ZQK44) was predicted using MaSIF-site. Then, the center of the interface was predicted by finding the patch with the highest mean interface score. Once the center of the interface was identified, the descriptor of this center point was matched to all patches in the 10954 proteins, for a total of 52 million fingerprints. Matches were ignored if the descriptor distance was greater than 1.7 or if the interface score was less than 0.9. The matches that passed this filter were explicitly aligned using our second stage alignment protocol. For this benchmark we used a simpler scoring function to rank each transformation, once correspondences were established between points (Supp. Fig 7), a score was computed according to the function $f = \sum_{ij} \frac{1}{d_{ij}^2}$ where d_{ij} is the descriptor distance between binder point i and target points j, such that i and j are within 1.0 Å. The top ten matches were then visually identified, showing the mouse PD1 (PDB id: 3BIK) as the top scoring match (ranked #1-#7), followed by the ground truth, wildtype match ranked #8.

Supplementary note 10. The precomputing time of the PDB files to generate surfaces with features and runtime for MaSIF-search and MaSIF-site neural networks is dependent on the protein size, and is thus plotted in Supp. Fig. 12. For example, a 125 amino acid protein is processed in 99.4 s accounting CPU, System and GPU times. GPU times were measured using 'wall-clock' time, since standard UNIX time tools do not account for GPU processing time. All times were measured on an Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz, and an NVIDIA Tesla K40 GPU running Red Hat Enterprise Linux 7.4. PDB files precomputations were performed on CPUs, while neural network calculations were performed on GPUs.

3 Fast end-to-end learning on protein surfaces

This chapter is a postprint version based on an article published in the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition in 2021 (Pages 15272-15281) in accordance with the publisher.

Authors
Freyr Sverrisson[1,2*], Jean Feydy[3*], Bruno E. Correia[1,2], Michael M. Bronstein[3,4]

[*] These authors contributed equally.

Affiliations
[1] Institute of Bioengineering, École Polytechnique Fédérale de Lausanne, [2] Swiss Institute of Bioinformatics, Lausanne, Switzerland, [3] Department of Computing, Imperial College London, London, UK, [4] Twitter, London, UK.

Author contributions
F.S., J.F., B.E.C and M.M.B. designed the overall method and approach. M.M.B. and B.E.C supervised the research. F.S. and J.F. designed and implemented the dMaSIF surface generation method. J.F. designed and implemented the dMaSIF convolutional operator. F.S. designed and implemented the network architectures. F.S. performed the experiments and benchmarks. F.S., J.F., M.M.B. and B.E.C wrote the manuscript. All authors read and commented the manuscript.

3.1 Abstract

Proteins' biological functions are defined by the geometric and chemical structure of their 3D molecular surfaces. Recent works have shown that geometric deep learning can be used on mesh-based representations of proteins to identify potential functional sites, such as binding targets for potential drugs. Unfortunately though, the use of meshes as the underlying representation for protein structure has multiple drawbacks including the need to pre-compute the input features and mesh connectivities. This becomes a bottleneck for many important tasks in protein science.

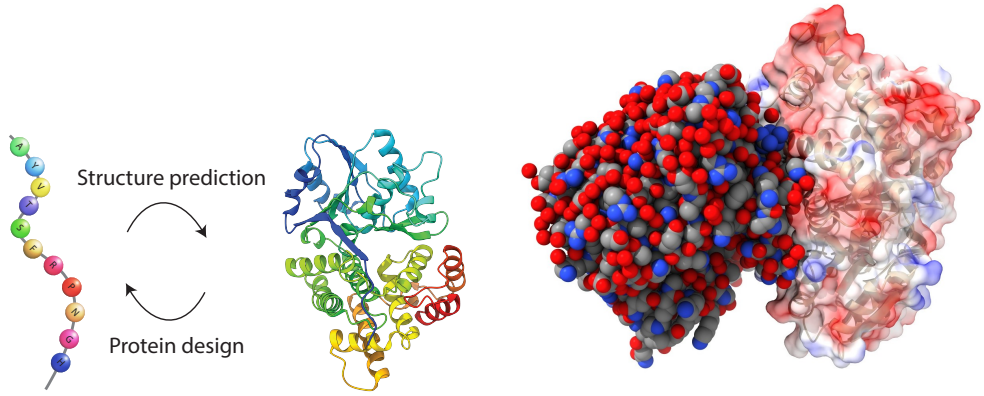
In this paper, we present a new framework for deep learning on protein structures that addresses these limitations. Among the key advantages of our method are the computation and sampling of the molecular surface on-the-fly from the underlying atomic point cloud and a novel efficient geometric convolutional layer. As a result, we are able to process large collections of proteins in an end-to-end fashion, taking as the sole input the raw 3D coordinates and chemical types of their atoms, eliminating the need for any hand-crafted pre-computed features.

To showcase the performance of our approach, we test it on two tasks in the field of protein structural bioinformatics: the identification of interaction sites and the prediction of protein-protein interactions. On both tasks, we achieve state-of-the-art performance with much faster run times and fewer parameters than previous models. These results will considerably ease the deployment of deep learning methods in protein science and open the door for end-to-end differentiable approaches in protein modeling tasks such as function prediction and design.

3.2 Introduction

Proteins are biomacromolecules central to all living organisms. Their function is a determining factor in health and disease, and being able to predict functional properties of proteins is of the utmost importance to developing novel drug therapies. From a chemical perspective, proteins are polymers composed of a sequence of amino acids (Fig. 3.1.a). This sequence determines the structural conformation (fold) of the protein, and the structure in turn determines its function. In a folded protein, hydrophobic (water-repelling) residues typically cluster within the core of the protein, while hydrophilic (water-attracting) residues are exposed to water solvent on its surface. The properties of this surface dictate the type and the strength of the interactions that a protein can have with other molecules (Fig. 3.1.b). Analysing this complex 3D object is therefore a fundamental problem in biology: models for protein structures can be used to understand the possible interactions between a protein and its environment, and consequently predict the functions of these macromolecules in living organisms.

Since proteins are predominant drug targets, the study of their interactions with other molecules is a key problem for fundamental biology and the pharmaceutical industry. Classical drugs are small molecules designed to bind to a protein of interest, with a binding site that usually



(a) Folding and design.

(b) Interacting surfaces.

Figure 3.1: Three major problems in structural biology. (a) Protein design is the inverse problem of structure prediction. (b) Two interacting proteins represented as an atomic point cloud (left) and as a molecular surface (right) that abstracts out the internal fold (shown semi-transparently). Protein surfaces display a number of geometric (e.g. concave and convex regions) and chemical (e.g. charges) features. Identifying their binding is a complex problem that can be addressed with geometric deep learning.

has noticeable 'pocket-like' structure. Targets with flat surfaces that exhibit no pockets have long been a challenge for drug developers and are often deemed 'undruggable'. The possibility of addressing such targets with specifically designed protein molecules (known as biological drugs or 'biologics') is a fast emerging field in drug-development holding the promise to provide novel therapeutic strategies for many important diseases (e.g. cancer, viral infections, ect.).

Deep learning methods have increasingly been applied to a broad range of problems in protein science [63], with the particularly notorious success of DeepMind's AlphaFold to predict 3D protein structure from sequence [149]. Recently, Gainza et al. [61] introduced MaSIF, one of the first conceptual approaches for *geometric deep learning* on protein molecular surfaces allowing to predict their binding. The main limitations of MaSIF stem from its reliance on pre-computed meshes and handcrafted features, as well as significant computational time and memory requirements.

Main contributions. In this paper, we present dMaSIF (differentiable molecular surface interaction fingerprinting), a new deep learning approach to identify interaction patterns on protein surfaces that addresses the key drawbacks of MaSIF. Our architecture is completely free of any precomputed features. It operates directly on the large set of atoms that compose the protein, generates a point cloud representation for the protein surface, learns task-specific geometric and chemical features on the surface point cloud and finally applies a new convolutional operator that approximates geodesic coordinates in the tangent space. All these

computations are performed on the fly, with a small memory footprint. Notably, we implement all core calculations as reductions of symbolic "distance-like" matrices, supported by the recent KeOps library [54] for PyTorch [124]: the high performance routines of this toolbox allow us to design a method which is fully differentiable and an order of magnitude faster and more memory efficient than MaSIF. This in turn allows us to make predictions on larger collections of protein structures than was previously practical, and opens the door to end-to-end protein optimization and *de novo* protein design using geometric deep learning.

3.3 Related works

Deep learning in protein science. Proteins can be represented in different ways, the 1D aminoacid sequence being the simplest and most abundant source of data. Recent methods have taken advantage of the wealth of protein sequences available in public databases and shown how unsupervised embeddings borrowed from the field of Natural Language Processing can improve function prediction [3, 21, 140]. Deep learning is also becoming a key component in many pipelines for protein folding (i.e. inferring the 3D structure from the aminoacid sequence) [5, 174, 149, 175]. Many of these pipelines predict pair-wise distances and other geometric relations between different residues and use these as constraints in later structural refinements. Protein design, which can be considered as 'inverse structure prediction' (i.e. predict a sequence that will fold into a particular structure), has also benefited from deep learning methods [76]. We refer to [63] for a comprehensive overview.

To model protein interactions, surface-based representations are especially attractive: they automatically abstract the less relevant internal parts of the protein fold, which do not contribute to the interaction. The Molecular Surface Interaction Fingerprinting (MaSIF) [61] method pioneered the use of mesh-based geometric deep learning to predict protein interactions. Its authors showed the application of MaSIF for classifying binding sites for small ligands, discriminating sites of protein-protein interaction in surfaces and predicting protein-protein complexes.

Nevertheless, in spite of its conceptual importance and impressive performance, the MaSIF method has significant drawbacks that limit its practical applications for protein prediction and design. First, it takes as inputs mesh-based representations of a protein surface, that must be generated from the raw atomic point cloud as a preprocessing step. Second, it relies on hand-crafted chemical and geometric features that must also be pre-computed and stored on the hard drive. Third, it uses MoNet [118] mesh convolutions on precomputed geodesic patches, which becomes prohibitively expensive in terms of memory and run time when working with more than a few thousand proteins.

Deep learning on surfaces and point clouds. Deep learning on non-Euclidean structured data such as meshes, graphs and point clouds, known under the umbrella term *geometric deep learning* [28], has recently become an important tool in computer vision and graphics.

Instead of considering geometric shapes as objects in a 3D Euclidean space and applying standard deep learning pipelines (e.g. based on 2D views [170], volumetric [153], space partitioning [137, 167, 158] and implicit representations [34]), geometric deep learning seeks to develop a non-Euclidean analogy of filtering and pooling operations. Boscaini et al. [113] proposed the first geometric CNN-like architecture (Geodesic CNN) based on intrinsic local charting on meshes. Follow-up works improved on these results using patch operators based on anisotropic diffusion (ACNN [27]), Gaussian mixtures (MoNet [118]), splines [53], graph message passing (FeastNet [165]), equivariant filters [131, 71], and primal-dual mesh operators [116]. We refer to [132] for a recent survey. Point clouds are often used as a native representation of 3D data coming from range sensors, and have recently gained popularity in computer vision in lieu of surface-based representations. First works on deep learning on point clouds were based on deep learning on sets [179] (PointNet [133] and PointNet++ [134]). DGCNN [168] uses graph neural networks [17] on kNN graphs constructed on the fly to capture the local structure of the point cloud. Additional tangent space [158] and volumetric [10] convolution operators were also considered, see a recent survey paper [69].

3.4 Our approach

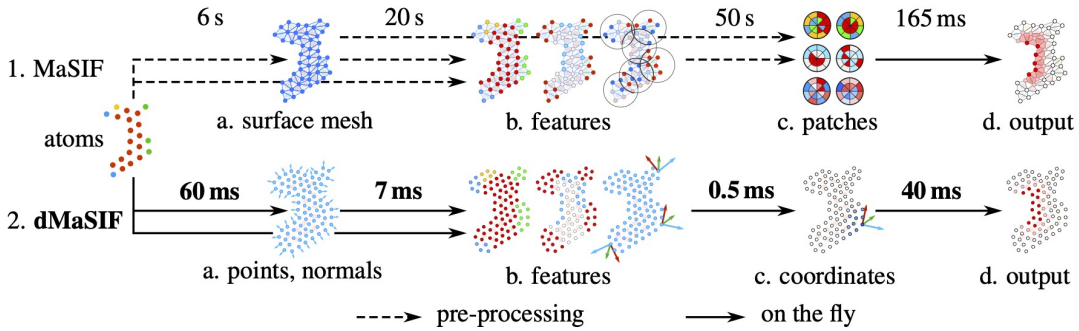


Figure 3.2: Both MaSIF and dMaSIF go through the same steps for interface prediction on protein surfaces. Starting from a raw atomic point cloud, we compute (a) a representation of the protein molecular surface, (b) geometric and chemical features, and (c) local coordinate systems; (d) a binding site is then predicted by a geometric convolutional neural network operating on (quasi-)geodesic patches on the protein surface. MaSIF precomputes steps (a)–(c), whereas we compute them on the fly 600 times faster. For every step, we display average run times per protein for inference on the site prediction task described in Section 3.5. Our method results in an accuracy level on par with MaSIF while alleviating the need for pre-calculations and providing significant speed-up for both inference and training.

Working with protein surfaces. In the following, we describe a new efficient end-to-end architecture for geometric deep learning on protein molecules. The premise of our work is that protein molecular surfaces carry important geometric and chemical information indicative of the way they interact with other molecules. Though we showcase our method on predicting binding properties (arguably, the most important task in structural biology and

drug design), it is generic and can be trained on other problems, and in principle, extended to other biomolecules.

Our method works on successive geometric representations of a protein, illustrated in Figure 3.2. The input is provided as a cloud of atoms $\{a_1, \dots, a_A\} \subset \mathbb{R}^3$, with chemical types in the list [C, H, O, N, S, Se] encoded as one-hot vectors $\{t_1, \dots, t_A\} \subset \mathbb{R}^6$. We then represent the surface of the protein as an oriented point cloud $\{x_1, \dots, x_N\} \subset \mathbb{R}^3$ with unit normals $n_1, \dots, n_N$ in $\mathbb{R}^3$. We associate feature vectors $f_1, \dots, f_N$ to these points and progressively update them by convolution-like operations; the dimension of these features varies from 16 (10 geometric + 6 chemical features as input) to 1 (binding score as output) throughout our network. Our data comes from the Protein Data Bank [19], with protein structures that are typically made up of A = 3K–15K atoms and molecule sizes in the range 30 Å–300 Å (one ångström is equal to 10^{-10} m); we sample their surfaces at a resolution of 1 Å to work with N = 6K–15K points at a time.

We stress that unlike most other works for surface processing, our method *does not* rely on mesh structures, kNN graphs, or space partitioning of any kind. We compute exact interactions between all points of a protein surface efficiently using the recent KeOps library [31, 54] for PyTorch [124] that optimizes a wide range of computations on generalized distance matrices. [I]

3.4.1 Surface generation

Fast sampling. The surface of a protein can be described as the level set of a smooth distance function or *meta ball* [22] (Figure 3.3a). To represent the six different atom types accurately, we associate an atomic radius σ_k to each atom a_k and define the smooth distance function as

$$\mathrm{SDF}(x) = -\sigma(x) \cdot \log \sum_{k=1}^{A} \exp\left(-\frac{\|x - a_k\|}{\sigma_k}\right), \tag{3.1}$$

for any $x \in \mathbb{R}^3$. With a stable log-sum-exp reduction and with

$$\sigma(x) = \frac{\sum_{k=1}^{A} \exp(-\|x - a_k\|)\, \sigma_k}{\sum_{k=1}^{A} \exp(-\|x - a_k\|)}, \tag{3.2}$$

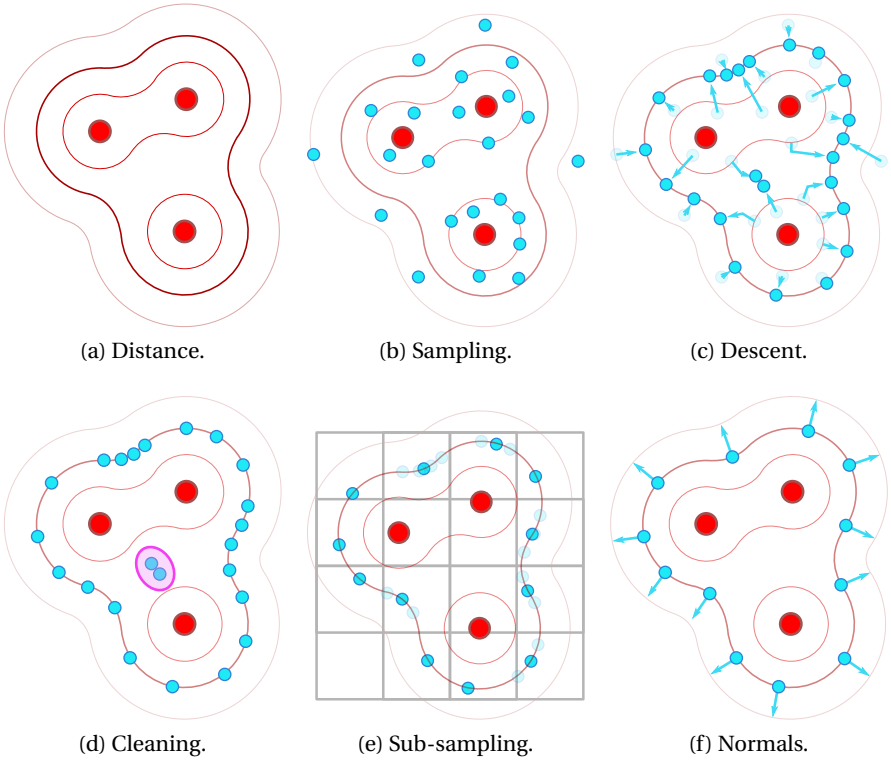
we have the average atom radius in a neighborhood of point x.

As shown in Figure 3.3b, we sample the level set surface at radius $r = 1.05$ Å by minimizing the squared loss function:

$$E(x_1, \dots, x_N) = \frac{1}{2} \sum_{i=1}^{N} (\mathrm{SDF}(x_i) - r)^2, \tag{3.3}$$

[I] The size 5K–20K and dimension 3 of our point clouds appear to be a sweetspot for KeOps in 'bruteforce mode', thanks to *contiguous* operations that stream much better on GPUs than the *scattered* memory accesses of graph-based and hierarchical methods.

on a random Gaussian sample. KeOps allows us to implement this sampling strategy efficiently on batches of more than 100 proteins at a time (see Figure 3.13a).

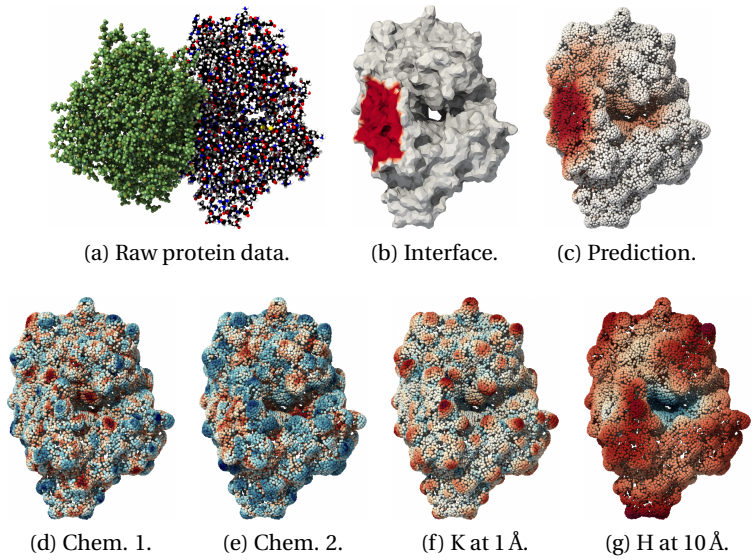


(a) Distance. (b) Sampling. (c) Descent.

(d) Cleaning. (e) Sub-sampling. (f) Normals.

Figure 3.3: Sampling algorithm for protein surfaces. (a) Given the input protein (encoded as an atomic point cloud $\mathbf{a}_1, \dots, \mathbf{a}_A$, in red), its molecular surface is represented as a level set of the smooth distance function (3.1) to the atom centers. (b) To sample this surface, we first generate a point cloud $\mathbf{x}_1, \dots, \mathbf{x}_{N=AB}$ in the neighborhood of our protein (in blue): for every atom center, we draw $B = 20$ points from $\mathcal{N}(\mu = \mathbf{a}_k, \sigma = 10\text{Å})$ and (c) let this random sample converge towards the target level set by gradient descent on (3.3) – we use 4 gradient steps with a learning rate of 1. (d) We then remove points trapped inside the protein: we keep a sample if the distance function at this location is close to our target value of $r = 1.05\,\text{Å}$ within a margin of 0.10 Å, and if making four consecutive steps of size 1 Å in the direction of the gradient of the distance function increases it by more than 0.5 Å. (e) We then put all points in cubic bins of side length 1 Å and keep one average sample per cell; this ensures that our sampling has uniform density. (f) Finally, the gradient of the distance function at location $\mathbf{x}_i$ is normalized to be used as a normal $\mathbf{n}_i$.

Descriptors. Point normals n_i are computed using the gradient of the distance function (3.1). To estimate a local coordinate system (n_i, u_i, v_i), we first smooth this vector field using a Gaussian kernel with $\sigma = 12\,\text{Å}$, i.e. use $n_i \leftarrow \text{Normalize}\left(\sum_{j=1}^{N} \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) n_j\right)$. We then compute tangent vectors u_i and v_i using the efficient formulae of [47]. Let $n_i = [x, y, z]$ be a



(a) Raw protein data. (b) Interface. (c) Prediction.

(d) Chem. 1. (e) Chem. 2. (f) K at 1 Å. (g) H at 10 Å.

Figure 3.4: Illustration on the binding of the `1OJ7` pair. (a) The Protein Data Bank documents interactions between proteins `1OJ7_D` (right) and `1OJ7_A` (left, green). Can we learn to predict this 3D binding configuration from the un-registered structures of both proteins? (b) MaSIF tackles this problem as a surface segmentation problem. The binding site (red) is the ground truth signal that MaSIF tries to predict from precomputed chemical and geometric features, such as the electrostatic potential. It relies on mesh convolutions on the preprocessed molecular surface of the protein. (c) dMaSIF predicts the binding site without using any precomputed mesh structure or features. We perform all computations on an oriented point cloud, generated from the raw atom coordinates as in Figure 3.3. Data-driven chemical features (d-e) as well as Gaussian (f) and mean (g) curvatures at different scales are computed on the fly and given as inputs to a fast convolutional architecture that we describe in Figure 3.5. Rendering done with ParaView [12].

unit vector, $s = \text{sign}(z)$, $a = -1/(s+z)$ and $b = a\,x\,y$, then

$$u_i = [\,1 + sax^2, sb, -sx\,], \qquad v_i = [\,b, s + ay^2, -y\,]. \tag{3.4}$$

For each point x_i, we then find the 16 nearest atom centers $\{a_1^i, \dots, a_{16}^i\}$ with types $\{t_1^i, \dots, t_{16}^i\}$ encoded as one-hot vectors in $\mathbb{R}^6$. We compute a vector of chemical features f_i in $\mathbb{R}^6$ by applying a Multi-Layer Perceptron (MLP) to the vectors $[t_k^i, 1/\|x_i - a_k^i\|]$ in $\mathbb{R}^7$, performing a summation over the indices $k = 1, \dots, 16$ and applying a second MLP to the result. As illustrated in Figure 3.6, using simple MLPs with a single hidden layer of dimension 12 is enough to learn rich chemical features, such as the Poisson-Boltzmann electrostatic potential.

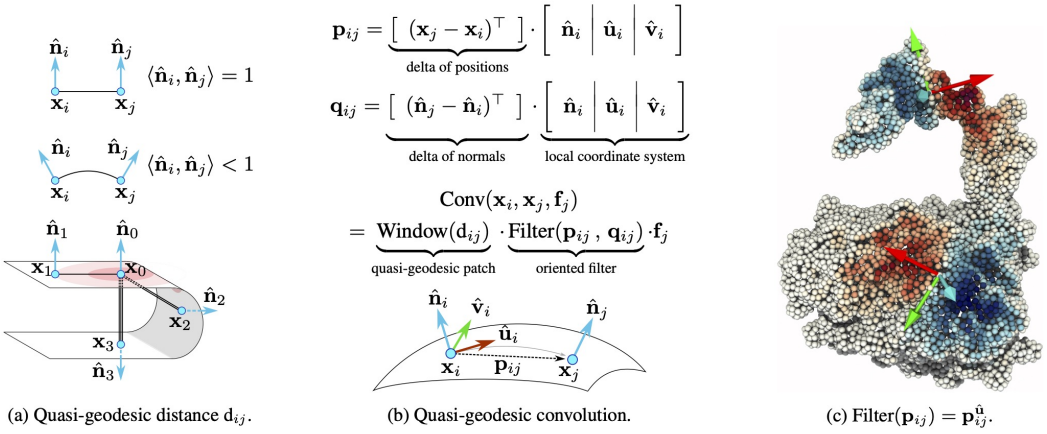
3.4.2 Quasi-geodesic convolutions on point clouds

Convolutions on 3D shapes. To update the feature vectors f_i and progressively learn to predict the binding site of a protein, we rely on (quasi-)geodesic convolutions on the molecular surface. This allows us to ensure that our model is fully invariant to 3D rotations and translations, takes decisions according to *local* chemical and geometric properties of the surface, and is not influenced by atoms located deep inside the volume of a protein. These modelling hypotheses hold for many protein interaction problems and prevent our network from overfitting on the few thousands of protein pairs that are present in our dataset.

In practice, geometric convolutional networks combine pointwise operations of the form $f'_i \leftarrow \text{MLP}(f_i)$ with local inter-point interactions of the form:

$$f'_i \leftarrow \sum_{j=1}^{N} \text{Conv}(x_i, x_j, f_j)\,, \tag{3.5}$$

where f_i and f'_i denote feature vectors associated to the point x_i and the $\text{Conv}(x_i, x_j, f_j)$ operator puts a trainable weight on the relationship between the points x_i and x_j. The sum can possibly be replaced by a maximum or any other reduction or *pooling* operation.



(a) Quasi-geodesic distance d_{ij}. (b) Quasi-geodesic convolution. (c) Filter$(\mathbf{p}_{ij}) = \mathbf{p}_{ij}^{\hat{\mathbf{u}}}$.

Figure 3.5: We use an approximation of the geodesic distance (3.6) to implement fast quasi-geodesic convolutions on oriented point clouds. (a) The weighted distance d_{ij} between points $\mathbf{x}_i$ and $\mathbf{x}_j$ is equal to $\|\mathbf{x}_i - \mathbf{x}_j\|$ if the unit normal vectors $\mathbf{n}_i$ and $\mathbf{n}_j$ point towards the same direction, but is larger otherwise. In this example, the points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ lay at equal distance of the reference point $\mathbf{x}_0$ in $\mathbb{R}^3$; but since the reference normal $\mathbf{n}_0$ is aligned with $\mathbf{n}_1$, orthogonal to $\mathbf{n}_2$ and opposite to $\mathbf{n}_3$, we have $d_{0,1} = \|\mathbf{x}_0 - \mathbf{x}_1\| < 2 \cdot d_{0,1} = d_{0,2} < 3 \cdot d_{0,1} = d_{0,3}$. (b) We leverage this behaviour to prevent information leakage "across the *volume*" of a protein. We combine a Gaussian window on the weighted distance d_{ij} with a parametric "Filter" to aggregate features $\mathbf{f}_j$ between neighbors on a protein *surface*. (c) Our formulae induce local coordinate systems that closely mimic the structure of genuine geodesic patches – defined here by a Gaussian window of deviation $\sigma = 10$ Å. On smooth surfaces, they enable the computation of "quasi-geodesic" convolutions at a much lower cost than mesh-based methods.

Working with oriented point clouds. Numerous methods have been proposed to mimic surface operators with convolution operators on meshes or point clouds – see Section **??** and especially [158, 109, 173, 159]. In this work, we leverage the *normal vectors* that are produced by our sampling algorithm to define a fast quasi-geodesic convolutional layer that works directly on oriented point clouds. The KeOps library lets us implement this operation efficiently, *without any offline precomputation* on the surface geometry.

As illustrated in Figure 3.5, we approximate the geodesic distance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ of a protein surface with unit normals $\mathbf{n}_i$ and $\mathbf{n}_j$ as:

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| \cdot (2 - \langle \mathbf{n}_i, \mathbf{n}_j \rangle) \tag{3.6}$$

and localize our filters using a smooth Gaussian window of radius $\sigma \in \{9, 12\}$ Å, $w(d_{ij}) = \exp(-d_{ij}^2 / 2\sigma^2)$. In the neighborhood of any point $\mathbf{x}_i$ of the surface, two 3D vectors then encode the relative position and orientation of neighbor points $\mathbf{x}_j$ in the local coordinate system $(\mathbf{n}_i, \mathbf{u}_i, \mathbf{v}_i)$:

$$\mathbf{p}_{ij} = [\,\mathbf{p}_{ij}^n, \mathbf{p}_{ij}^u, \mathbf{p}_{ij}^v\,], \qquad \mathbf{q}_{ij} = [\,\mathbf{q}_{ij}^n, \mathbf{q}_{ij}^u, \mathbf{q}_{ij}^v\,].$$

Different choices for the trainable "Filter" on these 3D vectors let us encode a wide range of operations. We focus here on polynomial functions and MLPs instead of the popular Mixture-of-Gaussian filters [118], but note that this choice has little impact on the expressive power of our model.

Local orientation, curvatures. We must stress, however, that the pair of tangent vectors $(\mathbf{u}_i, \mathbf{v}_i)$ orthogonal to the normal $\mathbf{n}_i$ is only defined up to a rotation in the tangent plane. To work around this problem at a low computational cost, we follow [115] and orient the first tangent vector $\mathbf{u}_i = \mathbf{u}(\mathbf{x}_i)$ along the geometric gradient $\nabla^{\mathbf{u},\mathbf{v}} P(\mathbf{x}_i)$ of a trainable potential $P(\mathbf{x}_i) = P_i = \mathrm{MLP}(\mathbf{f}_i)$, computed from the input features using a small MLP. We approximate its gradient using a derivative of Gaussian filter on the tangent plane, implemented as a quasi-geodesic convolution:

$$\nabla P(\mathbf{x}_i) \leftarrow \tfrac{1}{N} \sum_{j=1}^{N} w(d_{ij})\, [\,\mathbf{p}_{ij}^u, \mathbf{p}_{ij}^v\,]\, P_j \in \mathbb{R}^2 \tag{3.7}$$

and then update the tangent basis $(\mathbf{u}_i, \mathbf{v}_i)$ using standard trigonometric formulae

Local curvatures are computed in a similar fashion [30]. We use quasi-geodesic convolutions with Gaussian windows of radii σ that range from 1 Å to 10 Å and quadratic filter functions to estimate the local covariances $\mathrm{Cov}_{\sigma,i}^{\mathbf{u},\mathbf{v}}(\mathbf{p},\mathbf{p})$ and $\mathrm{Cov}_{\sigma,i}^{\mathbf{u},\mathbf{v}}(\mathbf{p},\mathbf{q})$ of the point positions and normals as 2×2 matrices in the tangent plane $(\mathbf{u}_i, \mathbf{v}_i)$. With $\lambda = 0.1$ Å a small regularization parameter, the 2×2 shape operator at point $\mathbf{x}_i$ and scale σ is then approximated as $\mathrm{S}_{\sigma,i} = (\lambda^2, \mathrm{Id}_{2\times 2} + \mathrm{Cov}_{\sigma,i}^{\mathbf{u},\mathbf{v}}(\mathbf{X},\mathbf{X}))^{-1} \mathrm{Cov}_{\sigma,i}^{\mathbf{u},\mathbf{v}}(\mathbf{p},\mathbf{q})$, which allows us to define the Gaussian $\mathrm{K}_{\sigma,i} = \det(\mathrm{S}_{\sigma,i})$ and mean

$\mathrm{H}_{\sigma,i} = \mathrm{trace}(\mathrm{S}_{\sigma,i})$ curvatures at scale σ.

Trainable convolutions. Finally, the main building block of our architecture is a quasi-geodesic convolution that relies on a trainable MLP to weigh features in a geodesic neighborhood of the local reference point $\mathbf{x}_i$. We turn a vector signal $\mathbf{f}_i \in \mathbb{R}^F$ into a vector signal $\mathbf{f}'_i \in \mathbb{R}^F$ with:

$$\mathbf{f}'_i \leftarrow \sum_{j=1}^{N} w(d_{ij})\mathrm{MLP}(\mathbf{x}_{ij})\mathbf{f}_j \tag{3.8}$$

where MLP is a neural network with 3 input units, $H = 8$ hidden units, ReLU non-linearity and $F = 16$ outputs.

3.4.3 End-to-end convolutional architecture

Overview. We chain together the operations introduced in the previous sections to create a fully differentiable pipeline for deep learning on protein surfaces, illustrated in Figure 3.2. As a brief summary:

- We sample surface points and normals as in Figure 3.3.
- We use the normals $\mathbf{n}_i$ to compute mean and Gaussian curvatures at 5 scales σ ranging from 1 Å to 10 Å.
- We compute chemical features on the protein surface as described in Section 3.4.1. Atom types and inverse distances to surface points are passed through a small MLP with 6 hidden units, ReLU non-linearity and batch normalization [77]. Contributions from the 16 nearest atoms to a surface point $\mathbf{x}_i$ are summed together, followed by a linear transformation to create a vector of 6 scalar features.
- We concatenate these chemical features to the 5 + 5 mean and Gaussian curvatures to create a full feature vector of size 16.
- We apply a small MLP on this vector to predict orientation scores P_i for each surface point. We then orient the local coordinates $(\mathbf{n}_i, \mathbf{u}_i, \mathbf{v}_i)$ according to (3.7).
- We apply successive trainable convolutions (3.8), MLPs and batch normalizations on the feature vectors $\mathbf{f}_i$. The numbers of layers, the radii of the Gaussian windows and the number of units for the MLPs are task-dependent and detailed in the Supplementary Material.
- As a final step for site identification, we apply an MLP to the output of the convolutions to produce the final site/non-site binary output. For interaction prediction, we compute dot products between the feature vectors of both proteins to use them as interaction scores between pairs of points.

Asymmetry between binding partners. When trying to predict binding interactions for protein pairs, we process both interacting proteins identically up to the convolutional step. We then introduce some asymmetry by passing each one of the two binding partners through a separate convolutional network. This allows the network to find *complementary* (instead of similar) regions on both surfaces, such as convex bulges and concave pockets. We note that MaSIF encoded such an asymmetry by inverting the sign of the precomputed features on one of the two surfaces.

3.5 Experimental Evaluation

Benchmarks. We test our method on two tasks introduced in [61]. The tasks come from the field of structural bioinformatics and deal with predicting how proteins interact with each other.
Binding site identification: we try to classify the surface of a given protein into interaction sites and non-interaction sites. Interaction sites are surface patches that are more likely to mediate interactions with other proteins: understanding their properties is a key problem for drug design and the study of protein interaction networks. The identification of the interaction site is unaware of the binding partner.
Interaction prediction: we take as inputs two surface patches, one from each protein involved in a complex, and predict if these locations are likely to come into close contact in the protein complex. This task is key to prediction tasks like protein docking, i.e. predicting the orientation of two proteins in a complex.

Dataset. The dataset comprises protein complexes gathered from the Protein Data Bank (PDB) [19]. We use the training / testing split of [61], which is based on sequence and structural similarity and was assembled to minimize the similarity between structures of the interfaces in the training and testing set. For site identification, the training and test sets include 2958 and 356 proteins, respectively; 10% of the training set is reserved for validation. For interaction prediction, the training and test sets include 4614 and 912 protein complexes, respectively, with 10% of the training set used for validation.

The average number of points used to represent a protein surface is $N = 11549 \pm 1853$ for our generated point clouds, compared to 6321 ± 1028 points for MaSIF.[II] Proteins are randomly rotated and centered to ensure that methods which rely on atomic point coordinates do not overfit on their spatial locations.

Baselines. Our main baselines are the MaSIF-site and MaSIF-search models [61]. For the MaSIF baselines, we use the pre-trained models and precomputed surface meshes and input

[II] This smaller sampling size of MaSIF stems from the large time and memory requirements of this method, which prohibits the use of finer meshes.

features provided by the authors. Additionally, in order to show the benefits of our convolutional layer, we benchmark it against PointNet++ [134] and Dynamic Graph CNN (DGCNN) [168], two popular state-of-the-art convolutional layers for point clouds.

Implementation. We implement our architectures with PyTorch [124] and use KeOps [54] for fast geometric computations. For data processing and batching, we use PyTorch Geometric [52]. For the PointNet++ and DGCNN baselines, we use PyTorch Geometric implementations – but rely on KeOps symbolic matrices to accelerate the construction of kNN graphs and thus guarantee a fair comparison. For the MaSIF baselines, we use the reference implementation of [61].[III] All models are trained on either a single NVIDIA GeForce RTX 2080 Ti GPU or a single Tesla V100. Run times and memory consumption are measured on a single Tesla V100.

3.5.1 Surface and input feature generation

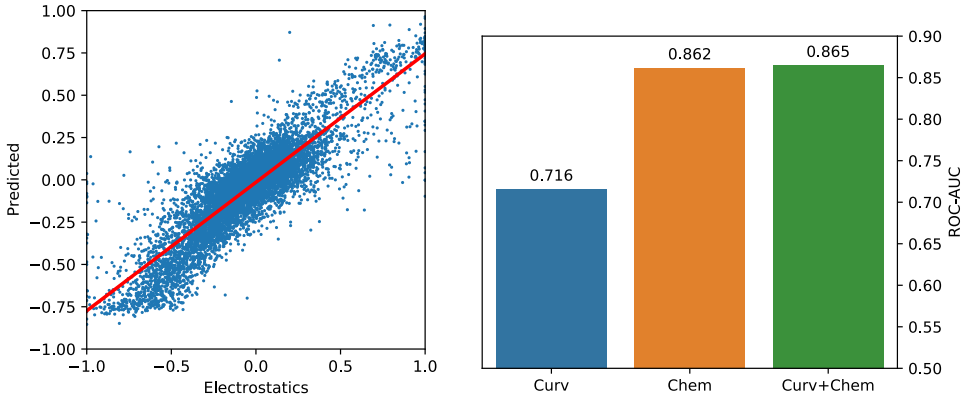
Precomputation. A key drawback of MaSIF is its reliance on the heavy precomputation of surface meshes and input features. These computations take a significant amount of time and generate large files that must be stored on disk. For reference, the pre-processed files used to train the MaSIF networks weigh more than 1TB. In sharp contrast, our method does not rely on any such pre-computation. Table 3.1 compares corresponding run times for both pipelines: our method is three orders of magnitude faster than MaSIF for these geometric computations.

Scalability. Our surface generation algorithm scales beneficially with an increasing batch size. In SM we show that the running time and memory requirement per protein of our method both decrease significantly when processing dozens of proteins at time the batch size. This is a consequence of the increased usage of the GPU cores and the smaller influence of fixed PyTorch and KeOps overheads.

Moreover, our method of surface generation makes it easy to experiment with different point cloud resolutions. Different tasks could benefit from higher or lower resolution and tuning it as a hyperparameter could have significant effects on performance. We show the effects of resolution on time an memory requirements in SM.

Quality of learned chemical features. Another notable drawback of MaSIF is its reliance on 'handcrafted' geometric and chemical features (Poisson-Boltzmann electrostatic potential, hydrogen bond potential and hydropathy) that must be precomputed and provided as input to the neural network. In contrast, we do not use any handcrafted descriptors and learn problem-specific features directly from the underlying atomic point cloud, provided as the sole input of our method. We argue that this information alone is sufficient to compute an informative

[III] Since MaSIF is implemented in TensorFlow [1], small discrepancies in measurements of memory consumption and running times are possible.



(a) Predicting electrostatics

(b) Ablation study.

Figure 3.6: Our network can compute chemical properties of the protein surface from the underlying atomic point cloud. (a) Predicted Poisson-Boltzman electrostatic potential vs. the ground truth. Correlation cofactor r=0.83 and RMSE=0.16. (b) Ablation study showing how chemical and geometric features affect the performance in predicting interaction sites (ROC-AUC).

Computation	MaSIF	Ours
Surface generation	6.11±6.18 s	59.0±15.2 ms*
Input features	19.69±16.08 s	6.59±1.22 ms*
Local coordinates	50.65±45.15 s	0.46±0.09 ms*

Table 3.1: Average "pre-processing" time per protein. Our method is about 1000 times faster than MaSIF and allows these computations to be performed on the fly, as opposed to the offline precomputations of MaSIF. *With batches of 128 proteins at a time.

chemical and geometric description of the protein surface. To support this statement, we show in Figure 3.6 the results of an experiment where our chemical feature extractor is used to regress the Poisson-Boltzmann electrostatic potential on surface points. The quality of our predicition suggests that our data-driven chemical features are of similar quality to the descriptors used by MaSIF – or better.

We also note the results of an ablation study for chemical and geometric features, depicted in Figure 3.6. They suggest that the concatenation of geometric curvatures to the vector of learned chemical features does not significantly improve the performance of the network for the site prediction task: we will investigate this point in future works.

3.5.2 Performance

Binding site identification. Results for the identification of binding sites are summarized in Figures 3.7–3.8b, which depict ROC curves and tradeoffs between accuracy, time and memory. We evaluate multiple versions of our architecture with varying numbers of convolution layers (1 vs 3) and patch sizes (5, 9, or 15Å). For comparison, we also show results when our convolutions

are replaced by DGCNN and PointNet++ architectures, all other things being equal.

A first remark is that if we use a single convolution layer with a Gaussian window of deviation $\sigma = 15$ Å, our method matches the best accuracy of 0.85 ROC-AUC produced by MaSIF – with 3 successive convolutional layers on patches of radius 9 Å. In this configuration, our network runs 10 times faster than MaSIF with an average time in the forward pass of 16 ms vs. 164 ms per protein. At the price of a modest increase of the model complexity (three convolution layers, and 36 ms on average per protein), we outperform MaSIF with a 0.87 ROC-AUC, detailed in Figure 3.7 (solid curves). Most remarkably, our models all have a small memory footprint (132 MB/protein), which is 11 times less than an equivalent MaSIF network (1492 MB/protein), 13 times less than DGCNN (1,681 MB/protein) and 30 times less than PointNet++ (3,995 MB/protein).

Interaction prediction. With a single convolutional layer architecture similar to that of MaSIF-search we reach a slightly higher performance of 0.82 vs. 0.81, as illustrated in Figure 3.7 (dashed). We remark that MaSIF-search reaches this level of accuracy using high dimensional feature vectors with 80 dimensions compared to our 16: understanding the influence of the number of convolutional "channels" on the performances of our network for different tasks will be an important direction for future works.

Note that MaSIF-search also relies on larger patches than MaSIF-site (12 Å vs. 9 Å), which causes a significant increase of run times to 727 ± 403 ms. On the other hand, our lightweight method runs in 17.5 ± 6.7 ms and is over 40 times faster at inference time.

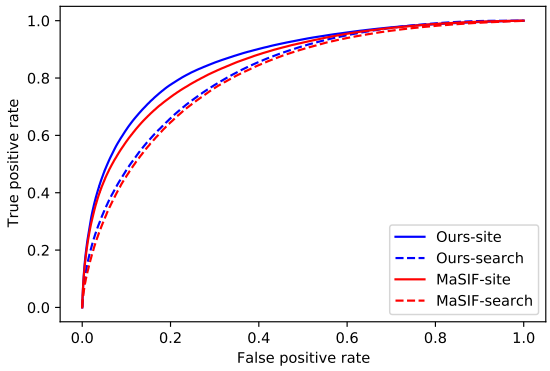
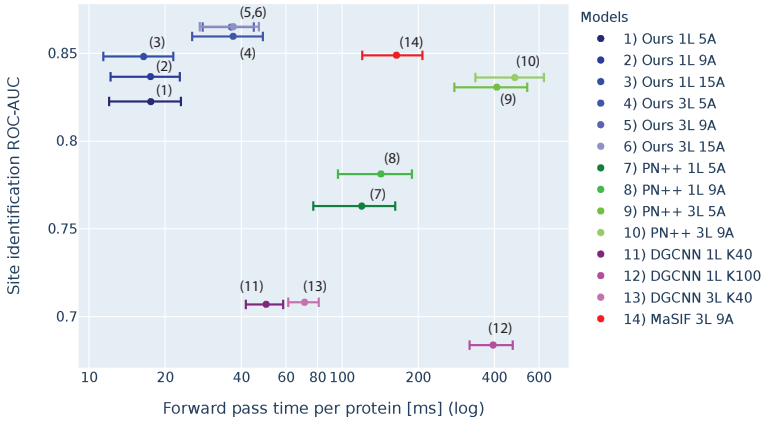


Figure 3.7: ROC curves comparing the performance of our method (blue) and MaSIF (red) on the task of binding site identification (solid curves) and search of binding partners (dashed). Our approach performs on par with MaSIF, achieving ROC-AUC of 0.87 (vs. 0.85) in site identification, and 0.82 (vs. 0.81) in identifying binding partners.



(a) Accuracy vs. Run time

(b) Accuracy vs. Memory footprint

Figure 3.8: (a) Accuracy (site identification ROC-AUC) vs. Run time (forward pass/protein in ms) of different architectures. Models are identified by the convolutional operator used, number of convolutional layers, and the value of σ used for the Gaussian window. PointNet++ models are identified by the radius of the neighborhood and DGCNN models by the number of nearest neighbours. (b) Accuracy (site identification ROC-AUC) vs. Memory footprint (MB/protein) of different architectures.

3.6 Conclusion

We have introduced a new geometric architecture for deep learning on protein surfaces, enabling the prediction of their interaction properties. Our method is an order of magnitude faster and more memory efficient than previous approaches, making it suitable for the analysis of large-scale datasets of protein structures: this opens the door to the analysis of entire protein-protein interaction networks in living organisms, comprising over 10K proteins.

The fact that our pipeline works on raw atomic coordinates and is fully differentiable makes it amenable to *generative* tasks, with the possibility of performing a true end-to-end design of new proteins for diverse biological functions, namely in terms of the design of binders for specific targets. This opens fascinating perspectives in drug design, including biologics for targeting disease relevant targets (e.g. cancer therapy, antiviral) that display flat interaction surfaces and are impossible to target with small molecules.

More broadly, we believe that our new algorithmic and architectural ideas for deep learning on 3D shapes through fast on-the-fly computations on point clouds will be of general interest to computer vision and graphics experts. Conversely, we hope that our work will draw the attention of this community to some of the most important and promising problems in structural biology and protein science.

3.7 Supplementary

3.7.1 Description of network architectures

A high level description of our networks for both site identification and interaction prediction can be found in Figs. 3.9a and 3.9b respectively. In these diagrams, "FC(I,O)" denotes a fully connected (linear) layer with I input channels and O output channels; "LR" denotes a Leaky ReLU activation function with a negative slope of 0.2; "BN" denotes a batch normalization layer. Red, blue and green blocks denote atom properties, surface descriptors and feature vectors, respectively.

We estimate chemical features on the generated surface points using the architecture described in Fig. 3.11. This module takes as inputs the atom coordinates and types, along with the surface point coordinates. For each point on the surface, the network finds the 16 nearest atoms and assigns a 6-dimensional chemical feature based on the atom types and their distances to the point. As detailed in Fig. 3.10, we concatenate these chemical features to a 10-dimensional vector of geometrical features, which approximate the mean and Gaussian curvatures at different scales.

We then pass these input feature vectors through a sequence of convolutional layers (Fig. **??**). As discussed in Section 3 of the paper, we first use the surface normals $\mathbf{n}_i$ to build local tangent coordinate systems and orient the unit tangent vectors $\mathbf{u}_i$, $\mathbf{v}_i$ according to the gradient of an orientation score P_i. Finally, we use this complete description of the surface geometry to establish quasi-geodesic convolutional windows and progressively update our feature vectors.

The DGCNN and PointNet++ baselines replace the "convolutional" block of our architecture with standard alternatives provided by PyTorch Geometric. We keep the same numbers of channels as for our method (8 for the site prediction task, 16 for the search predicition task) and benchmark runs with several interaction radii and number of K-nearest neighbors.

3.7.2 Description of the training process

We filter the datasets according to the criteria described in [61]. To be considered in our benchmarks, each protein must have at least 30 interface points and the interface has to cover less than 75% of the total surface area.

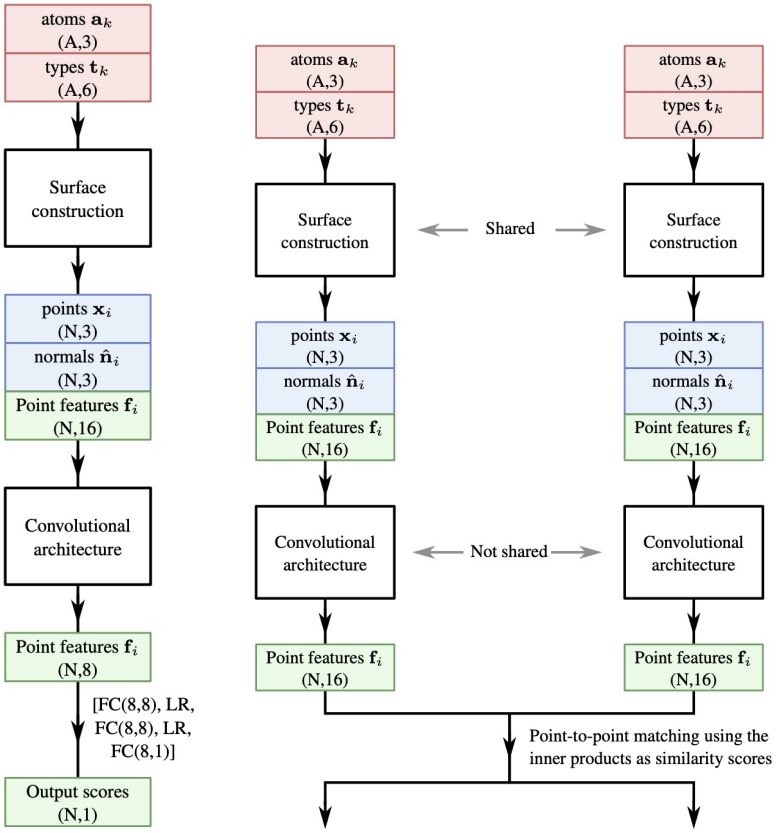
Binding site identification. We detail our hyperparameters in Table 3.2. Surfaces are generated in batches, but predictions are only performed on single proteins at a time. From each protein, 16 positives and 16 negatives locations are randomly sampled and the loss function is computed on these points. We found that this process stabilized the training process and improved generalization. Labels are mapped from precomputed MaSIF meshes by finding the nearest neighbours. Furthermore, if a point is further than 2.0Å away from any precomputed mesh point, it is labeled as non-interface. The loss is computed as the binary cross entropy

Parameter	Site	Search
Optimizer	AMSGrad	AMSGrad
Learning rate	3×10^{-4}	3×10^{-4}
Epochs	50	100
Descriptor dimensionality	8	16
Early stopping	Yes	Yes

Table 3.2: Hyperparameters for our training loops.

between the labels and the predictions.

Interaction prediction. Surface generation and prediction are performed in the same way as for binding site identification. However, as detailed at the end of Section 3.3 in the paper, each binding partner is passed through a separate convolutional network. The prediction scores are then computed by taking the inner product between the convolutional embeddings of the two proteins. Pairs of points are labeled as interacting if they are less than 1Å from each other. From each protein, 16 positives and 16 negatives were randomly sampled. The loss was computed as the binary cross entropy.



(a) Site prediction. (b) Search prediction.

Figure 3.9: a) Overview of our architecture for the site prediction task, that we handle as a binary classification problem of the surface points. The "surface construction" block is detailed in Figure 3.10, while the "convolutional architecture" is detailed in Figure **??**. b) Overview of our architecture for the search prediction task.

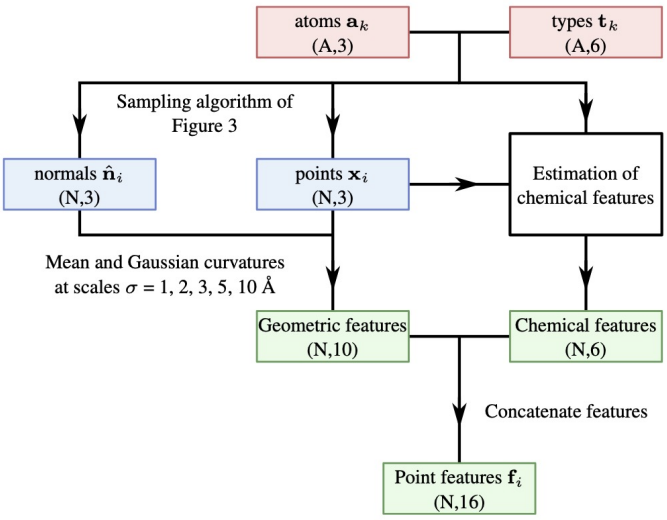


Figure 3.10: Construction of a surface representation, detailed in Section 3.1 of the paper. The "chemical features" block is detailed in Figure 3.11.

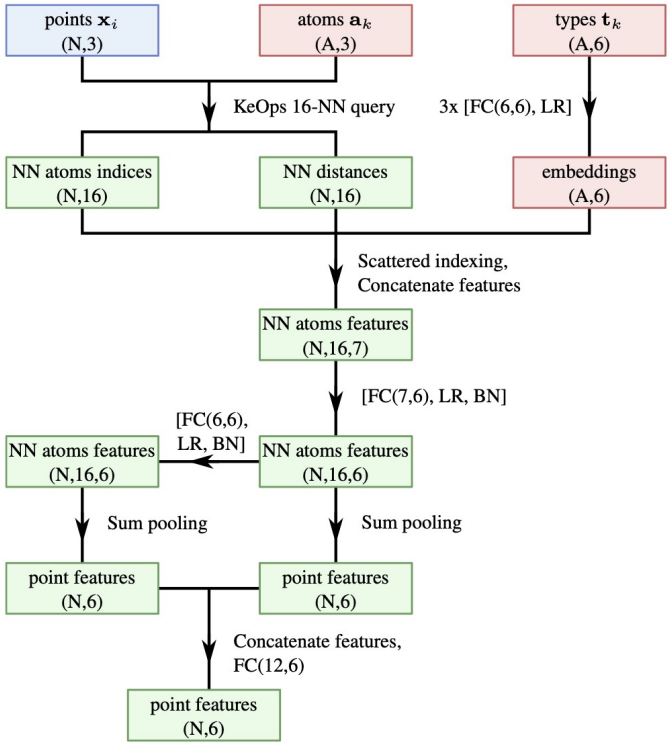
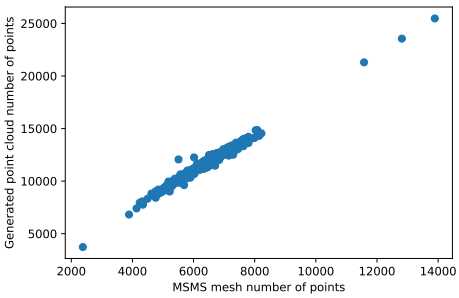
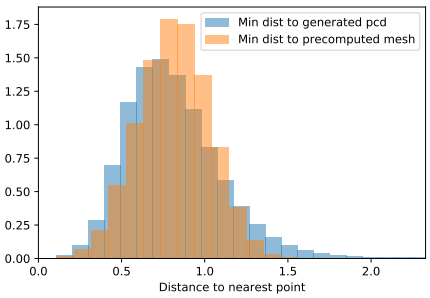


Figure 3.11: Estimation of chemical features from the raw atom types and coordinates.

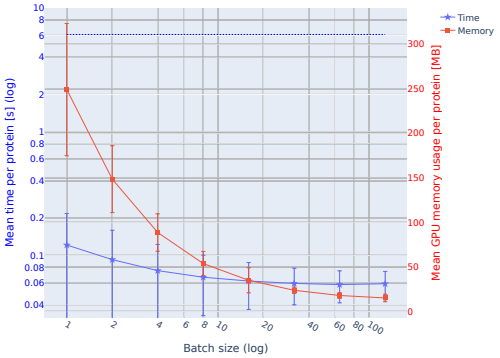


(a) Number of points.

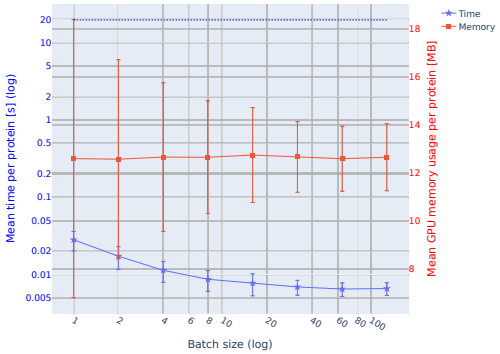


(b) Distances to closest points.

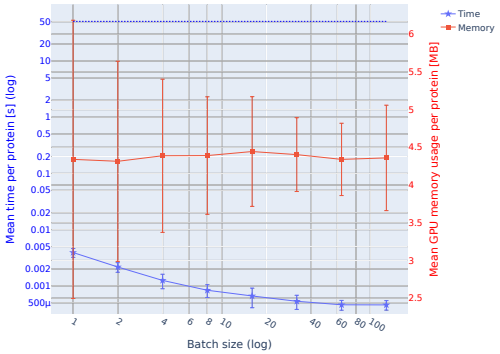
Figure 3.12: Quality control for our surface generation algorithm. (a) Number of points generated per protein by our method, as a function of number of points in the precomputed mesh used by MaSIF. As expected, we observe a nearly perfect linear correlation. (b) For each point generated by our method, we display in orange the distance to the closest point on the precomputed mesh. Conversely, we display in blue the histogram of distances to the closest generated point, for points on the MaSIF "ground truth" mesh. We noticed that the blue curve showed a very long tail (not visible on this figure). This comes from an artifact in the surface generation algorithm of MaSIF, which cuts out parts of proteins that have missing densities. We solved this discrepancy by removing these points from our dataset as well, and only display point-to-point distances in the 99th percentile – i.e. we treat the largest 1% distances as outliers, not displayed here.



(a) Surface generation.

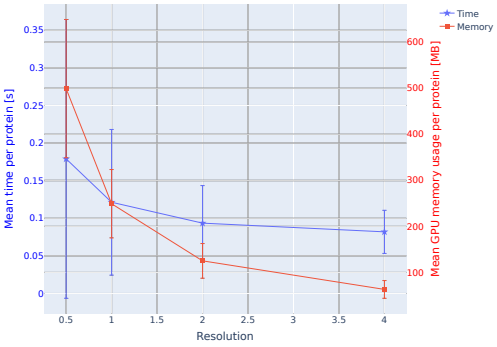


(b) Input features.

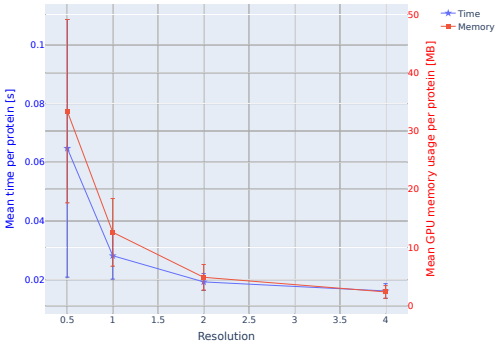


(c) Local coordinates.

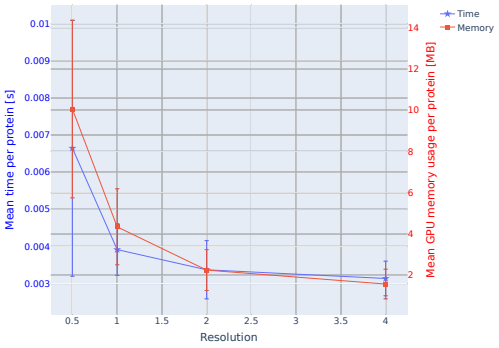
Figure 3.13: Computational cost of our "pre-processing" routines as functions of the batch size. We show the average time (blue curve and left axis, log scale) and memory (red curve, right axis, log scale) requirements of our method per protein, as a function of the number of proteins that are processed in parallel by our implementation. The dotted blue line shows the average time used by MaSIF to generate a surface mesh from the same atomic point cloud.



(a) Surface generation.

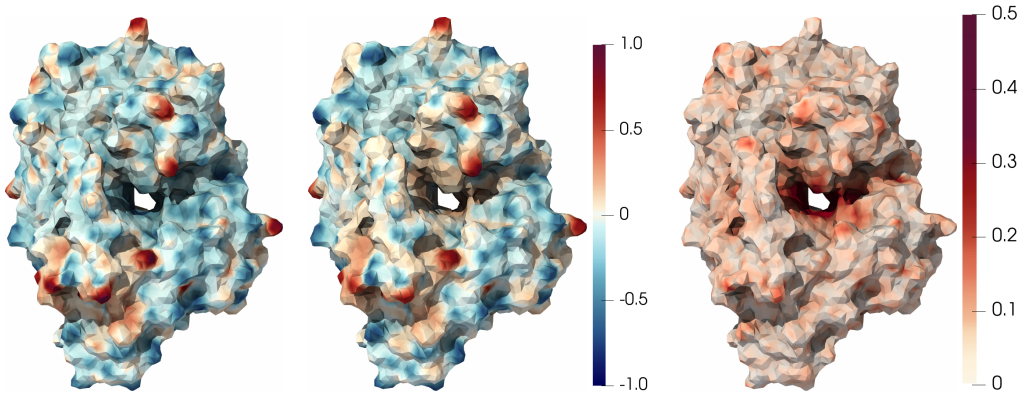


(b) Input features.



(c) Local coordinates.

Figure 3.14: Computational cost of our "pre-processing" routines, as a function of the sampling resolution. We display the time (blue line and blue axis) and memory (red line and red axis) requirements of the pre-convolutional steps of our architecture as a function of the resolution of the generated point cloud. As expected, increasing the sampling density of our surface generation algorithm (i.e. using a lower resolution) results in longer processing times.



(a) Ground truth. (b) Prediction. (c) Error.

Figure 3.15: Additional rendering, illustrating the results of Figure 7 of the paper on the 10J7_D protein from the Protein Data Bank. We display the ground truth (a) and predicted (b) electrostatic potential on the protein surface. The error (c) is small, with RMSE=0.14. We note that most of the error is located inside the cavity.

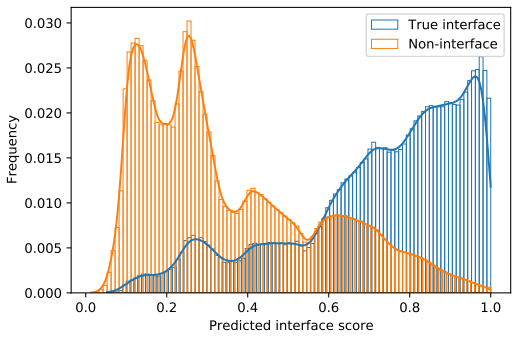


Figure 3.16: Additional display for the site prediction task. We display the distributions of predicted interface scores for both true interface points (blue) and non-interface points (orange). The separation is clear, resulting in a ROC-AUC of 0.87 in Figure 8 of the paper.

4 DiffMaSIF: Score-Based Diffusion Models for the Docking of Protein Surfaces

This chapter draws from our ongoing efforts to adapt our earlier techniques for learning on protein surfaces into a fully trainable docking network. We demonstrate that diffusion models based on surfaces surpass their residue-based counterparts in performance. Additionally, we juxtapose our approach with conventional docking methods. The findings showcased in this thesis represent preliminary evaluation outcomes, primarily due to the substantial computational demands. We are presently gathering evaluations from the complete testing set of our dataset and conducting a more comprehensive analysis of our method's failure modes in anticipation of manuscript preparation.

Preliminary author list
Freyr Sverrisson[1,2*], Mehmet Akdel[3*], Dylan Abramson[3*], Alex Goncearenco[3], Yusuf Adeshina[3], Zachary Carpenter[3], Luca Naef[3], Michael M. Bronstein[3,4], Bruno E. Correia[1,2]

[*] These authors contributed equally.

Affiliations
[1] Institute of Bioengineering, École Polytechnique Fédérale de Lausanne, [2] Swiss Institute of Bioinformatics, Lausanne, Switzerland, [3] VantAI, New York, NY 10036, [4] University of Oxford, UK.

Author contributions
F.S., M.A. and D.A. designed and implemented the overall method and approach. Z.C., L.N., M.M.B. and B.E.C supervised the research. M.A. designed and implemented the data splitting method. A.G. performed the benchmarking against classical docking methods. Y.A. implemented metric evaluation.

4.1 Abstract

Predicting protein-protein complexes is one of the central challenges of computational structural biology. Inspired by the success of recent generative ML methods in small-molecule docking, we present DiffMaSIF, a score-based diffusion model for rigid protein-protein docking. The critical factor for protein interactions is the complementarity found within the physical surfaces of protein interfaces. Unlike previous ML methods which were confined to residue representations, DiffMaSIF advances the field by leveraging a surface-based molecular representation. This information is then integrated into an equivariant network, thereby efficiently addressing the task at hand. We further identify and rectify structural leakage in a commonly utilized training dataset, and establish new splits for the purposes of benchmarking DiffMaSIF. Our results demonstrate that DiffMaSIF not only outperforms contemporary ML methods in rigid protein docking, but also matches traditional docking tools at considerably low numbers of generated decoys.

4.2 Introduction

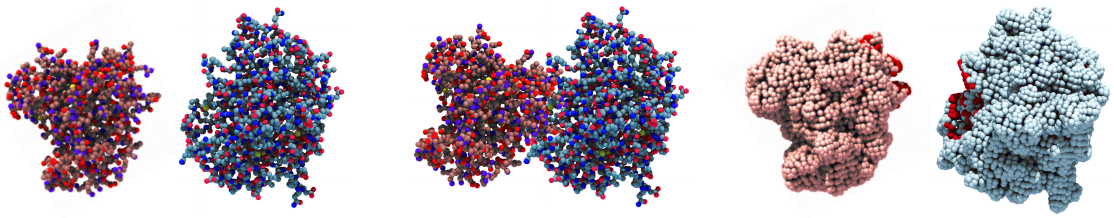
Proteins serve a multitude of functions within living organisms. Most of these functions are derived from how proteins interact with other molecules, which can be other proteins, other types of biological macromolecules, or small molecules.

The structure of a protein defines the kinds of interactions it can participate in. Computational methods for predicting protein structure have dramatically improved with the adaption of deep learning methods to the point where most protein structures can be predicted to near-experimental accuracy using the underlying protein sequence and information about its evolutionary history.

While these methods have considerably advanced the prediction of single-chain protein structures, the accurate prediction of multi-chain protein-protein complexes remains an ongoing challenge. The complexity of the task is dramatically increased by the fact that proteins are not structurally rigid. The rigidity varies over the structure where some parts are highly flexible (such as in loop regions) and others (such as helices) being more stable. This non-rigidity of proteins allows them to undergo structural rearrangement, called induced-fit, when coming into contact with another molecule. The data on such rearrangements are sparse compared to the wealth of possibilities which might prove to be a bottleneck for deep learning methods that rely too heavily on sequence homology and co-evolution.

Traditional methods for protein-protein docking typically involve constructing a pseudo energy function derived from physical principles along with the analysis of empirical protein-protein complexes, followed by the use of blackbox stochastic optimization techniques to search for minima within these energy functions. The search space of all possible conformations is infeasible to search exhaustively which is why rigid-body docking is typically the first step in traditional docking tools, followed by an iterative refinement, which takes into account

the flexibility of the molecules.



(a) Binder. (b) Target. (c) Docking configuration. (d) Binder and target surfaces.

Figure 4.1: **The rigid docking problem.** Let us consider a pair of proteins, the moving binder (a) and the fixed target (b). Our goal is to predict a rigid-body transformation of the binder that corresponds to a docking configuration (c) that has been observed in the wild. To this end, we build upon our previous works on protein surface fingerprints [61, 155]

For rigid-body protein docking, many traditional methods have taken advantage of the fact that protein complexes are characterized by high shape and chemical complementarity at their interface. Rigid-body docking was revolutionized by [84] by the usage of implicit representations of protein surfaces and by using fast Fourier transform of a correlation function to assesses the degree of shape complementarity and penetration upon rotation and translation of the molecules in three dimensions.

The surface representation of proteins has proven to be effective in predicting protein interactions using deep learning methods [61, 155]. The surface representation moreover offers a possible unified representation between small molecules and proteins, which can be advantageous in particular applications. Molecular glue degraders for example, a promising class of pharmaceuticals, are believed to modify the surface of a target protein such that natural, low-affinity interaction propensities are strengthened [95]. Modelling such modifications directly at the surface level could allow for generalization beyond atom-based models.

In this work, we introduce DiffMaSIF, a score-based diffusion model for rigid-body docking that emphasizes the surface representation of proteins. To promote generalization to both protein and small-molecule design, our model is deliberately limited to relying on structural features, rather than sequence homology or co-evolutionary features that have previously been advantageous for the task.

4.3 Background

4.3.1 Protein-Protein Docking

Predicting the three-dimensional structure of protein complexes has been one of the central problems of structural biology. Experimental methods such as X-ray crystallography provide us either with the structure of a protein in complex with its binding partner or in isolation by itself. The problem of taking a pre-determined protein complex, pulling the individual

protein chains apart, and re-assembling them is referred to as bound protein-protein docking whereas if each subunit has been characterized by itself it is referred to as unbound docking.

Traditional docking tools have typically divided the problem up into two stages. The first stage is a rigid-body docking stage where the 6-dimensional space of translations and rotations is thoroughly searched on a grid around the receptor. The second stage is a fine-tuning stage where the most plausible conformations from the previous stage are used as starting points for a more fine-grained search where side-chain and backbone movements are considered.

The surface shape-complementarity of two proteins bound to each other is a well-documented phenomenon [101, 81] and has been widely used in classical methods for protein-protein docking [58, 33, 48, 37]. Traditional methods in most cases start their rigid-body search by defining implicit representations of the protein surfaces which can be enriched with features such as electrostatics. They then take advantage of the fast Fourier transform (FFT) and Fourier correlation theory to rapidly scan the translational space. The rotational space on the other hand is near-exhaustively sampled.

EquiDock [62] was proposed as a deep learning method for rigid-body protein-protein docking. EquiDock forms a graph representation of the residues in the interacting proteins, and through SE(3)-equivariant operations predicts keypoints to align using the Kabsch algorithm. EquiBind [154] was later developed as an extension to predict the docking of small molecules to proteins.

Both AlphaFold-multimer and EquiDock are trained in a supervised fashion as opposed to being generative models that can be sampled from. DiffDock [40], a diffusion generative model for docking ligands to proteins learns a denoising process over the ridgid-body translation and rotation of the small molecule, along with its torsional degrees of freedom. DiffDock-PP [87] a derived method for predicting rigid-body protein-protein docking and operates on the residue graph of the two interacting proteins.

Our proposed method shares many similarities with DiffDock-PP in the sense that it is a learned denoising process on the space of translational and rotational degrees of freedom. It differs however from DiffDock-PP in the sense that we focus on the surface representation of the proteins and we focus on structural features rather than sequence homology or co-evolution which is incorporated into DiffDock-PP in the form of sequence embeddings derived from a large language model.

4.3.2 Score-based Diffusion Models

Score-based diffusion models integrate techniques from both score-based generative models and diffusion models into a unified framework. In score-based generative modeling, the score function $\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$ represents gradients of the data log-density $p(\mathbf{x})$. The score can be estimated via denoising score matching on noise-corrupted samples, without needing to compute intractable normalizing constants. Langevin dynamics can then sample from the estimated score model. Diffusion models perturb data $\mathbf{x}_0$ through Markov chains of added

Gaussian noise to obtain $\mathbf{x}_t$ at noise level t. The forward diffusion process can be represented as a stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x} = f(t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \tag{4.1}$$

where $f(t)$ and $g(t)$ represent drift and diffusion coefficients respectively, and $\mathrm{d}\mathbf{w}$ is Gaussian noise. The reverse process is modeled by learning an approximate conditional distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Score-based diffusion models leverage score functions to parameterize the generative diffusion process. The forward SDE incrementally adds noise to the data distribution $p_0(\mathbf{x})$. Critically, the reverse-time SDE is:

$$\mathrm{d}\mathbf{x} = [f(t) - g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \tag{4.2}$$

The score functions $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ can be estimated by a time-dependent score-based model $\mathbf{s}_\theta(\mathbf{x}, t)$ trained via score matching. This results in an estimated reverse SDE that can be numerically solved to sample from $p_0(\mathbf{x})$. Alternatively, the estimated reverse SDE can be converted to a probability flow ODE, enabling exact likelihood computation.

The ability to perform tractable likelihood evaluation is a major benefit, as the probability flow ODE enables exact likelihood computation. This allows the model to be quantitatively evaluated, in contrast to other generative models like GANs where the likelihood is intractable. Additionally, the score-based component can leverage arbitrary neural network architectures suitable for the data, such as convolutional networks for images. Architectural advances like U-Nets can be readily incorporated, providing modeling flexibility. Efficient sampling techniques like Langevin MCMC can be used to generate high-quality samples from the estimated reverse SDE, with the sampling process automatically concentrating on high-density regions. The score perspective also enables straightforward control over generated samples for tasks like class-conditional generation and image inpainting. The conditioning information can be flexibly incorporated into the score-based model component, enabling controllable generation. Furthermore, the diffusion process allows training from incomplete data by marginalizing out missing inputs. This is useful for applications like image inpainting where parts of the data are unobserved, allowing learning from partial data. In summary, score-based diffusion models combine complementary techniques from both score-based modeling and diffusion processes. This enables leveraging the advantages of both approaches – flexible likelihood-based modeling from score matching, and efficient Markov chain-based sampling from diffusion models. The result is a highly flexible framework achieving state-of-the-art results in generative tasks.

4.3.3 Deep learning on Protein Surfaces

We rely on methods from dMaSIF [155] to both generate protein surfaces and corresponding scalar features. In [61] the method MaSIF was developed to learn protein surface descriptors that could be used for predicting interaction properties. It was shown that MaSIF generalized

better than homology-based methods to predict binding properties of designed interaction sites. In [60] it was more over shown that MaSIF could be used to design *de novo* protein interactions.

dMaSIF was developed as a much faster alternative to MaSIF. It is a framework that incorporates a fully-differentiable method to generate protein surfaces and a fast convolutional operator that operates in the quasi-geodesic space of the surface.

4.4 Data

The DIPS dataset [161] has been widely used to benchmark deep learning methods for protein-protein interaction prediction, both for predicting pairs of interacting residues and for protein-protein docking. DIPS was originally split such that the testing set was composed of complexes from the Docking Benchmark 5 (DB5) [166] and the training set was composed of complexes from the PDB such that no protein had more than 30% sequence homology to any protein in the DB5.

There are a few possible issues with such a split: First, a single protein can have multiple distinct binding sites and the structural similarity between the interfaces formed at these sites can be very low. Second, some proteins, such as antibodies, might have high sequence similarity at a global level but at the binding site be very different from each other. Lastly, in some cases sequence similarity might not be sufficient do discriminate between structurally similar proteins.

In [62] the authors further partitioned the training dataset from DIPS into training, validation and testing sets based on protein family labels which they used to assess the performance of their method on rigid-body docking. They similarly partitioned the DB5 dataset and fine-tuned their rigid-body model on unbound structures from the training set of DB5 to assess performance in unbound docking. DiffDock-PP was trained and tested using the same splits for rigid-body docking but had not been evaluated on the unbound DB5 dataset.

We examined the quality of the rigid-body dataset split through an unbiased structure based leakage analysis. We performed all-vs-all structural alignments of complete chains and focused on the interface sites using FoldSeek [86]. Residues were classified as interface residues based on an 8Å alpha carbon distance threshold between interacting chains. The interactions below a minimum of 6 residues are filtered out. Binding site clusters were assigned from alignments that exhibited over 75% interface coverage. Pairs of these clusters were then used to define paired interface clusters. To assess the quality of the splits, we investigated each complex from the testing set, identified their corresponding paired interface clusters, and verified if these clusters also encompassed any PPI pairs from the training set. Our analysis indicated a severe data leakage in DIPS splits with 82% of the testing pairs clustered with training set members (Fig. 4.2).

In light of these discoveries, we opted to develop new dataset splits intended for rigid-body docking by employing the same structural interface clustering approach that revealed the data leakage issue. To construct a de-leaked benchmark dataset, we retrieved all protein complexes from the PDB, also incorporating recently deposited and lower resolution structures.

After structural clustering we randomly selected 15% of the clusters which contained 10 or more members, to compose the testing and validation sets. For the training dataset we picked the remaining clusters and did not perform any quality-based filtering.

4.5 Methods

4.5.1 Diffusion Process

Our study adopts a diffusion process approach, akin to the methodologies presented in [40, 87]. We focus on a combined space termed the product manifold, denoted as $\mathbb{P}$. This manifold is a combination of:

1. **3D Translation Group ($\mathbb{T}(3)$)**: Essentially, this is the space of all possible 3D translations, equivalently represented as $\mathbb{R}^3$.

2. **3D Rotation Group ($SO(3)$)**: This encapsulates all conceivable 3D rotations.

Translations: For any translational movement in the 3D space, we employ the equation:

$$d\mathbf{x} = \sqrt{d\sigma_{tr}^2(t)/dt}\, d\mathbf{w}$$

Where: - $d\mathbf{x}$ signifies the change in position. - $\sigma_{tr}^2(t)$ is the diffusing variance at a specific time t. - $d\mathbf{w}$ represents the 3D Brownian motion, a type of random motion in three dimensions.

Rotations: For the rotational aspect, the process is twofold:

1. We initially select a random axis, represented as $\hat{\omega}$, and a random angle ω constrained between 0 and π. 2. The likelihood of opting for a particular angle ω is expressed by:

$$p(\omega) = \frac{1-\cos\omega}{\pi} f(\omega)$$

Where $f(\omega)$ is a truncated series expression, as detailed in [102].

By distinctly defining the diffusion for $\mathbb{T}(3)$ and $SO(3)$, we can train a model to match the scores for each kind of movement. During the sampling phase, we amalgamate samples from both the translational and rotational processes. This involves a random rotation of the ligand around its centroid and a random translation. This integrated methodology facilitates the creation of a generative model for docking within the product manifold $\mathbb{P}$.

4.5.2 Model Architecture

The DiffMaSIF architecture is divided into two primary components: an encoder and a decoder. The encoder combines residue and surface-level representations to produce surface level node embeddings, complete with their coordinates and normals. Meanwhile, the decoder employs DGCNN (vector neuron) layers and an E(3)-equivariant graph convolution layer to forecast translation and rotation scores.

Input features

DiffMaSIF accepts residue and atom level features from both ligand and receptor proteins. We utilize pretrained Gearnet embeddings [181] as scalar input features for residues, complemented by their coordinates. Atom level features include a one-hot encoding of atom types and their coordinates. These are fed into a dMaSIF layer, which generates surface normals, positions, and scalar embeddings using dMaSIF's geodesic convolution layer. Subsequently, both Gearnet features and dMaSIF scalar embeddings are scaled to matching dimensions via MLP layers.

Encoder

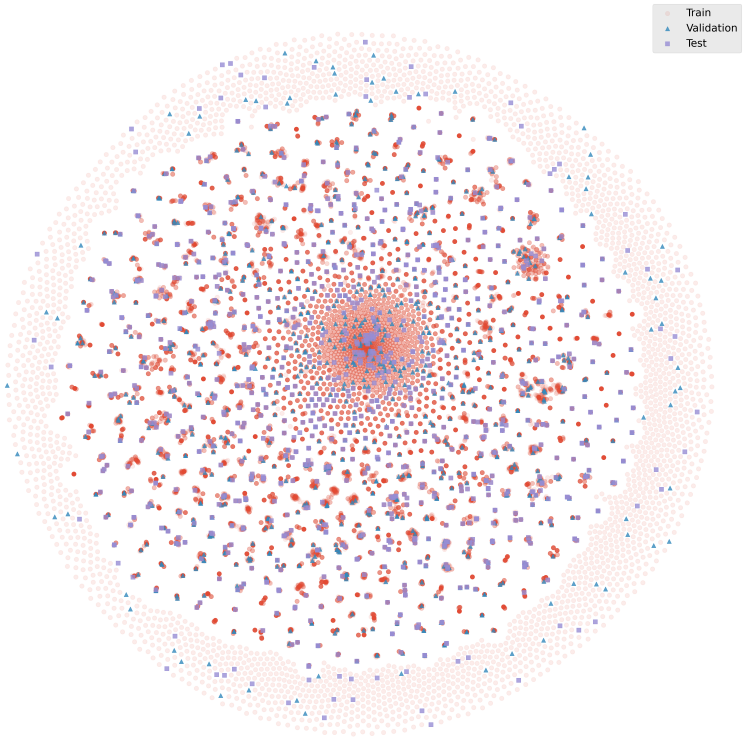
The DiffMaSIF encoder operates at the individual protein level, aggregating residue information onto the surface. To achieve this, DiffMaSIF constructs a heterograph consisting of residue nodes and surface nodes. Edges within this heterograph are formed among residue nodes and surface nodes based on a distance threshold. Subsequently, inter-node edges are established between residue and surface nodes in a similar manner. Edge embeddings are generated using a Gaussian function. We then apply graph neural network (GNN) node and edge convolutions within the node types (termed the intra-convolution layer) and from residue nodes to surface nodes (the inter-convolution layer).

Following the residue-to-surface message passing, we acquire scalar node features linked to their original coordinates and normals. These scalar features are directed to a SagPooling layer [105], which conducts graph-attention message passing and global self-attention to predict scores for each point. We rank these scores to select the top 512 nodes. At this juncture, we have distinct node sets for the ligand and receptor. We then form edges among ligand surface nodes (intra-edges) using k-nearest neighbors and establish edges between the ligand and receptor graphs (cross-edges). The node embeddings are finally merged with time embeddings generated by a sinusoidal position function.

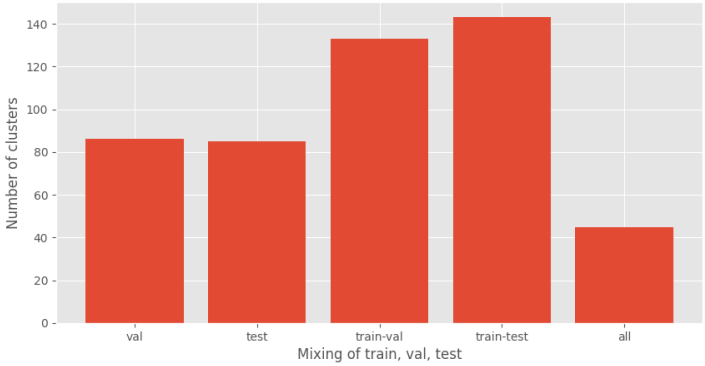
Decoder

Up to this stage, the encoder has processed and produced the ligand and receptor graphs separately, with their unique layers. In contrast, the decoder addresses the entire PPI graph as

a whole. The decoder's initial component is a DGCNN (vector neuron layers) [168, 45], which ingests coordinates and normal vectors, outputting higher-dimensional vector embeddings. These embeddings, along with surface positions and scalar features, are then fed to an E(3)-equivariant graph convolution layer [67]. The decoder's final output predicts ligand coordinate adjustments, offering a prediction of translation and rotation scores. The synergy of DGCNN and E(3)-equivariant graph convolutions ensures adherence to the geometric constraints of the protein surface structure.



(a)



(b)

Figure 4.2: a) Shows a 2-dimensional t-SNE [163] map of the results from the all-vs-all FoldSeek structural alignment where the structures have been color coded according to the original DIPS train-val-test split. The figure shows that structures of high structural similarity often end up in different sets which causes data-leakage. b) Shows the number of validation and testing clusters that do not have any data leakage (val and test in figure) compared to clusters that have a mix of structures from the training, validation and test sets.

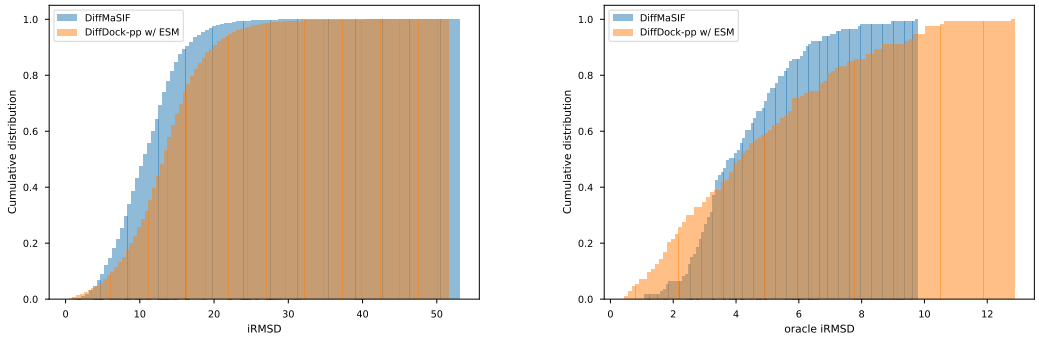
4.6 Results

4.6.1 Comparison to DiffDock-PP

To benchmark our approach, we retrained DiffDock-PP on our novel data splits, after having ensured reproducibility of results previously reported using the original splits. Fig. 4.3 illustrates the comparison of interface-RMSD (iRMSD) values between our method and the original DiffDock-PP. Additional metrics can be found in Supplementary Figures 4.10 and 4.11.

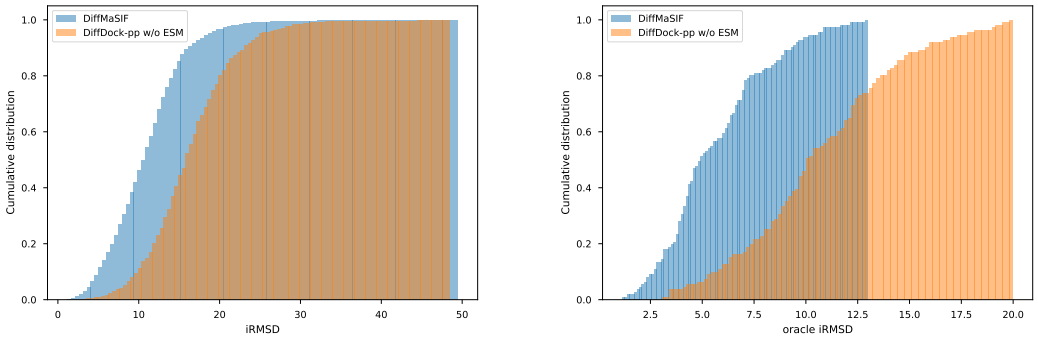
For each method we generated 40 poses per complex with a reverse ODE using 40 steps. Across all generated poses, as depicted in Fig. 4.3a, DiffMaSIF consistently exhibits a lower iRMSD compared to DiffDock-PP, with medians of 10.7Å and 13.4Å respectively. However, when selecting the optimal pose (assuming a flawless scoring function, an oracle) from a set of 40 generated for each complex (see Fig. 4.3b), DiffDock-PP displays a marginally extended tail towards reduced iRMSDs. Despite this, DiffMaSIF maintains a superior median iRMSD, registering 3.8Å against DiffDock-PP's 4.2Å. The variation in the median oracle iRMSD based on the number of generated decoys is detailed in Fig. 4.5.

DiffDock-PP incorporates ESM2 sequence embeddings [110] as input features, potentially capturing evolutionary information. Given that our methodology is grounded solely on structural data, we were intrigued to ascertain the extent of DiffDock-PP's reliance on these sequence embeddings. Consequently, we retrained DiffDock-PP, excluding the ESM2 embeddings. The comparative analysis with this model is presented in Fig. 4.4. The performance disparity between the two techniques is notably pronounced, with DiffMaSIF outperforming in both the entirety of generated poses and the top selections from the 40 generated.



(a) (b)

Figure 4.3: Performance comparison to DiffDock-PP over 113 structures from the testing set. a) Shows the cumulative distribution of all generated poses for the testing complexes for given values of iRMSD. b) Shows the cumulative distribution over the best (lowest iRMSD) generated complexes for each protein-protein pair.



(a) (b)

Figure 4.4: Performance comparison to DiffDock-PP which has been trained without using ESM2 embeddings over 111 structures from the testing set. a) Shows the cumulative distribution of all generated poses for the testing complexes for given values of iRMSD. b) Shows the cumulative distribution over the best (lowest iRMSD) generated complexes for each protein-protein pair.

4.6.2 Comparison with Conventional Docking Tools

We subsequently evaluated our approach against traditional rigid body PPI docking techniques, specifically PatchDock [48] and FRODock [64], using the test subset of our dataset. The protein chains were categorized into receptors and ligands, followed by re-docking. Both PatchDock and FRODock were executed using their default settings, producing a set of 40 poses ranked intrinsically by the respective methods. Each pose was then juxtaposed with the reference pose and evaluated based on three metrics: iRMSD (interface RMSD), lRMSD (ligand RMSD), and DockQ.

The comparative iRMSD results for these tools, juxtaposed with our method, are presented in Fig. 4.6. Additional metrics can be consulted in Supplementary Figures 4.10 and 4.11. Across all generated poses, DiffMaSIF consistently showcases a superior iRMSD, recording a median of 10.8Å in contrast to FRODock's 13.8Å and PatchDock's 14.3Å. In terms of oracle metrics however, DiffMaSIF registers a median of 3.5Å, while FRODock and PatchDock report medians of 1.9Å and 4.5Å, respectively. It's imperative to highlight that, despite our extraction of only 40 poses from both docking tools, FRODock and PatchDock internally generate a vast array of decoys. These decoys undergo ranking via a scoring function, followed by clustering based on similarity. The final output comprises structures derived from the centroids of these clusters. Given that our approach lacks such intricate scoring and decoy clustering, it is somewhat at a comparative disadvantage.

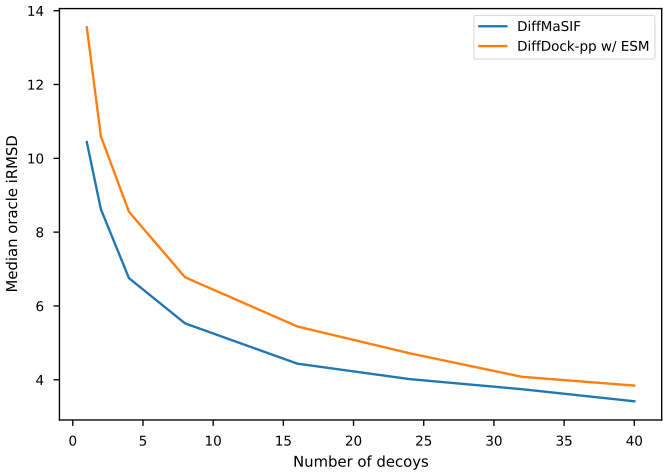
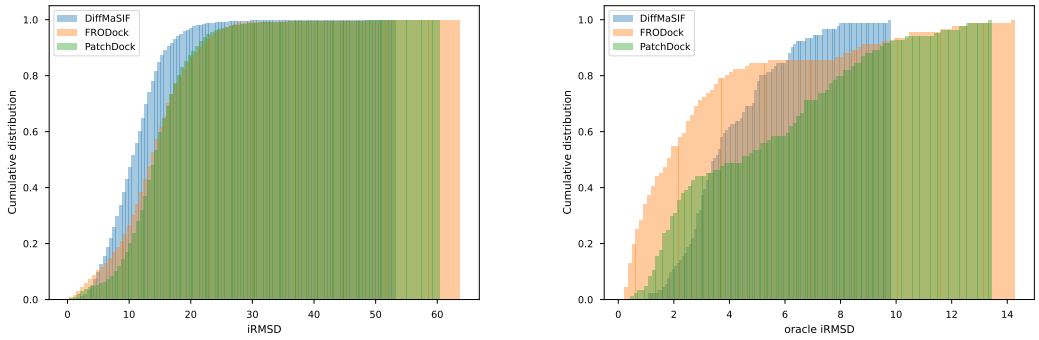


Figure 4.5: Median oracle iRMSD over 113 complexes from the testing set as a function of the number of generated poses per complex.



(a) (b)

Figure 4.6: Performance comparison to FRODock and PatchDock over 91 structures from the testing set. a) Shows the cumulative distribution of all generated poses for the testing complexes for given values of iRMSD. b) Shows the cumulative distribution over the best (lowest iRMSD) generated complexes for each protein-protein pair.

4.7 Conclusion

In this work, we have presented DiffMaSIF, a novel score-based diffusion model for rigid protein-protein docking that leverages protein surface representations. Our preliminary results demonstrate that DiffMaSIF surpasses contemporary machine learning techniques for this task. Compared to DiffDock-PP, DiffMaSIF achieves lower interface RMSD values across both the entirety of generated poses and the top selections. Notably, DiffDock-PP relies heavily on sequence embeddings, and its performance deteriorates significantly without them. In

contrast, our approach is grounded solely in structural data. Benchmarking against established rigid docking tools highlights DiffMaSIF's competitiveness despite its lack of intricate scoring and clustering procedures. Across all poses, DiffMaSIF reports superior interface RMSDs relative to PatchDock and FRODock. In terms of top selections, DiffMaSIF nears the median oracle RMSDs of these traditional techniques while generating far fewer internal decoys. While promising, these initial outcomes warrant more comprehensive evaluation over the complete test set and analysis of failure modes as future work. The preliminary findings presented herein demonstrate that surface-based representations hold strong potential to advance machine learning techniques for molecular docking.

4.8 Supplementary

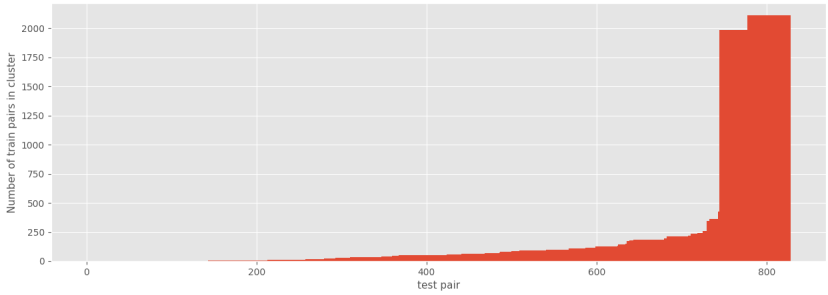


Figure 4.7: Number of training complexes in the same cluster as the testing complex.

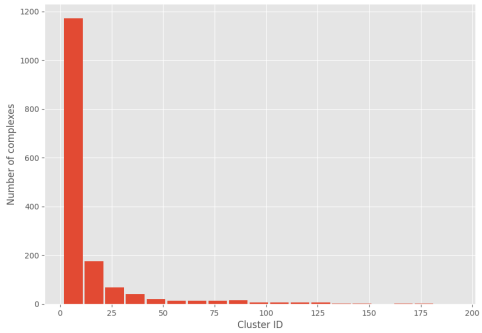
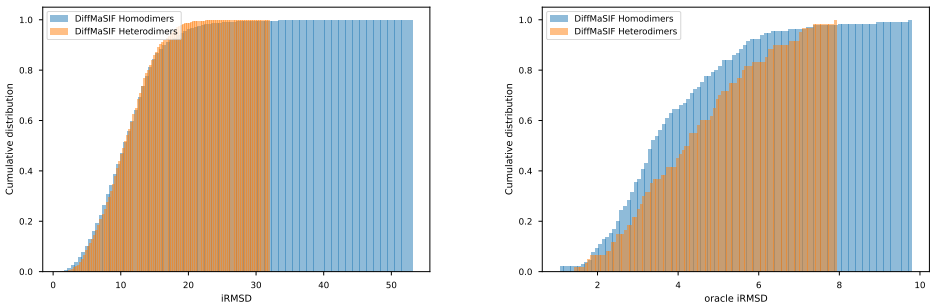
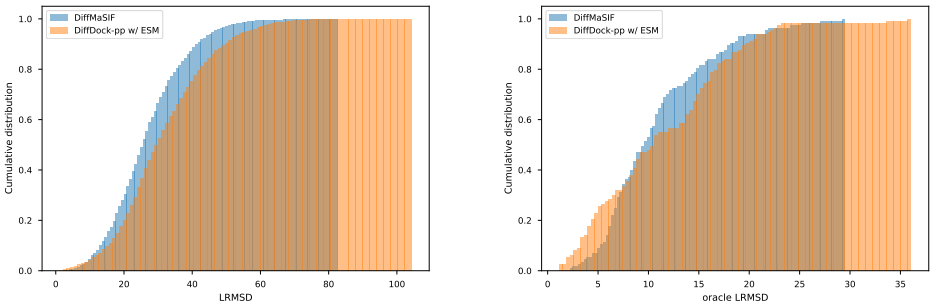


Figure 4.8: Number of complexes in each cluster.



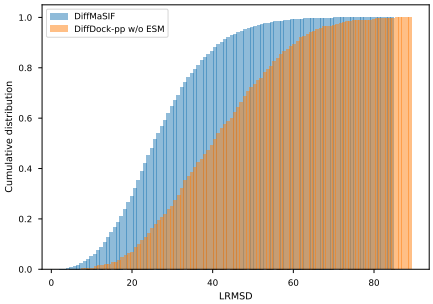
(a) (b)

Figure 4.9: Performance of our method on homo- vs. heterodimers

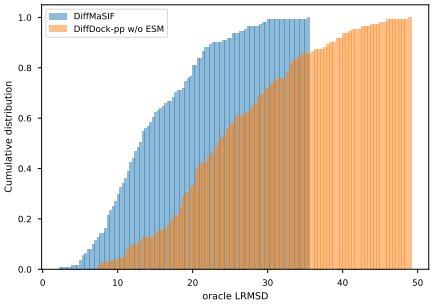


(a)

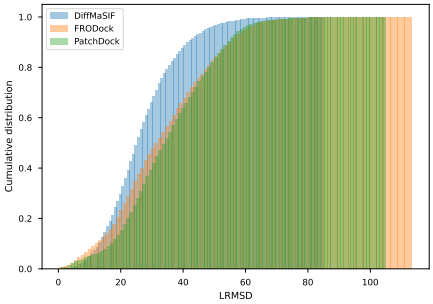
(b)



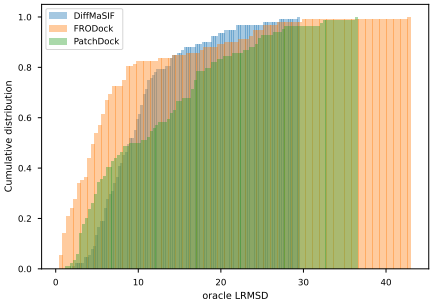
(c)



(d)



(e)



(f)

Figure 4.10: Ligand RMSD performance comparisons

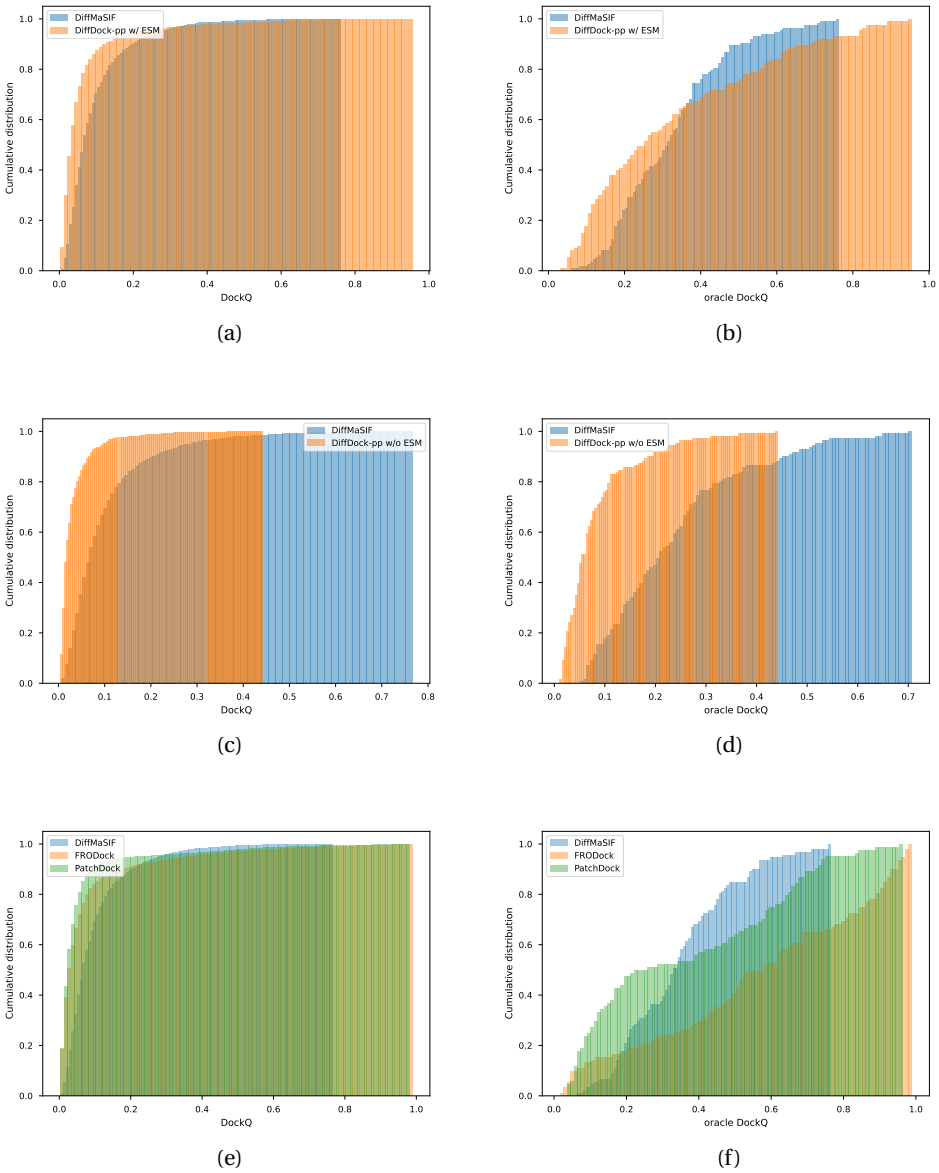


Figure 4.11: DockQ performance comparisons

5 Conclusions & Perspectives

Deciphering the intricate language of protein interactions is central to unraveling the complexities of biological systems. This thesis has introduced innovative computational methodologies for predicting protein interactions directly from structural data, independently of evolutionary history. By leveraging the power of deep learning techniques, we have made significant strides in understanding the complexities of protein interactions, which is central to understanding the intricacies of biological systems.

5.1 Summary of Main Findings

In Chapter 2, we presented MaSIF, a conceptual framework that establishes the viability of applying geometric deep learning to extract interaction fingerprints from molecular surfaces. By transforming surface patches into powerful numerical descriptors using neural networks in geodesic space, MaSIF extracts meaningful patterns correlated with diverse interaction types. Molecular surfaces provide a higher-level representation of protein structure, capturing patterns of chemical and geometric features indicative of binding modes and interactions. We showcased MaSIF's versatility across pocket-ligand prediction, protein-protein interaction site labeling, and ultrafast rigid docking searches. The results highlighted MaSIF's capacity to uncover predictive surface fingerprints relying solely on geometric and chemical properties, irrespective of evolutionary history. As proof-of-concept demonstrations, MaSIF was applied to diverse prediction challenges, including classifying ligand binding sites, identifying protein interfaces, and rapidly scanning for binding partners. The MaSIF-ligand model could accurately classify pockets based solely on surface features, even distinguishing highly similar ligands like NAD and NADP.

Chapter 3 introduced dMaSIF, uplifting MaSIF to enable end-to-end learning directly from atomic coordinates. dMaSIF constructs molecular surface representations on-the-fly using differentiable sampling. It computes geometric and chemical properties through small neural networks, eliminating hand-crafted features. We further implemented a novel convolution operator that establishes neighborhoods via approximated geodesic distances. Operating

directly on atomic coordinates, dMaSIF eliminates pre-processing bottlenecks and matches the accuracy of MaSIF while being orders of magnitude faster. This efficiency enables analyzing large structural datasets, previously infeasible. Chemical properties like electrostatic potential were accurately learned from just atom types and positions. By enabling differentiability, dMaSIF opens up new possibilities for geometry-based generative modeling in protein science.

In Chapter 4, we adapted our surface-based techniques into DiffMaSIF, a score-based diffusion model for protein-protein docking. DiffMaSIF integrates surface and residue-level information through an equivariant architecture to predict rigid binding configurations. Our results demonstrated DiffMaSIF's superior capacity over other machine learning techniques, matching established docking tools with far fewer generated decoys. The findings highlighted the benefits of surface-based modeling for capturing intermolecular complementarity.

Collectively, this thesis makes significant headway in demonstrating the potential of deep learning on molecular surfaces for predicting protein interactions. By elucidating interaction fingerprints, this thesis lays the foundations to move beyond naturally evolved proteins and rationally design novel interactions. The advancements presented herein are hoped to accelerate progress at the crossroads of biology, medicine, and artificial intelligence, pushing the boundaries of what we previously thought possible in the realm of protein science.

5.2 Broader Impacts

The techniques and methodologies developed in this thesis underscore the tremendous potential of applying advanced computational methods to further our understanding of protein structures and interactions. These approaches, rooted in the intersection of deep learning and biology, have the potential to accelerate discovery and innovation across a multitude of domains.

In the realm of drug design, surface-based interaction predictors can significantly aid the design of novel therapeutics. By uncovering druggable sites, predicting specificity, and enabling rapid virtual screening, these methods can revolutionize the way we approach drug discovery. This is particularly pertinent in our ongoing quest to address various diseases and health challenges.

Biotechnology stands to gain immensely from these advancements. Fast and differentiable generative models open new avenues for engineering proteins and biomolecules with desired functions. This could lead to the creation of novel proteins and bio-nanomaterials not limited by the constraints of natural evolution, potentially revolutionizing fields from sustainable energy to medical diagnostics.

In the sphere of basic biology, elucidating interaction mechanisms through interpretable geometric learning can unravel fundamental biomolecular processes. Such insights can provide a deeper understanding of life at the molecular level, shedding light on the intricacies

of cellular functions, regulation, and more.

The healthcare sector can benefit from an improved understanding of pathogenic protein interactions, which can inform diagnostic and therapeutic strategies. As we grapple with global health challenges, these computational tools could be pivotal in developing new treatments and interventions.

Furthermore, our methods advance the state-of-the-art in structural bioinformatics, benefiting myriad prediction pipelines. This could accelerate biological research across many domains, from scientists studying specific protein systems to gain insights into binding mechanisms, regulation, and downstream effects, to drug discovery where these methods could aid in identifying new therapeutic targets and designing novel protein therapeutics.

Beyond the immediate applications in biology and healthcare, the broader field of machine learning also stands to gain. The conceptual and algorithmic innovations around geometric deep learning and molecular generation have broad applicability, pushing the boundaries of what we understand about neural networks, data representation, and learning paradigms.

However, with great power comes great responsibility. The ability to rapidly generate in silico protein models could carry risks if misused, echoing concerns associated with DNA synthesis technologies. Proper governance frameworks will be essential to ensure responsible use. As we continue to push the boundaries of what's possible with computational biology, it's imperative that we approach these advancements with caution, ensuring that they are used ethically and responsibly.

Moreover, advancing computational structural biology contributes to the growth of artificial intelligence and its integration with the natural sciences. As deep learning matures as a scientific field, its collaboration with disciplines like biology creates synergy and opens new capabilities on both sides. This interdisciplinary spirit often seeds the most groundbreaking innovations.

By bringing together structural biology and artificial intelligence, this thesis exemplifies the potential of interdisciplinary research. The years ahead promise ever closer integrations between the computing and biological sciences, opening new frontiers in our understanding of life and the universe we inhabit.

5.3 Future Outlook

The advancements presented in this thesis, while significant, represent just the beginning of what promises to be a transformative journey in the realm of protein science. As we look to the future, several avenues beckon exploration, promising to further refine our understanding and capabilities in predicting protein interactions.

One of the most immediate directions is the expansion of the structural dataset. By covering a

greater diversity of proteins and complexes, we can improve the generalizability of our models across various protein folds and interaction types. High-throughput crystallization techniques and advancements in cryo-EM will be invaluable in generating more data. Additionally, computational models like AlphaFold present an exciting opportunity to supplement experimental data with high-quality predictions.

A deeper understanding of protein interactions necessitates that we move beyond static structures. Testing on unbound protein conformations is critical to better reflect in vivo conditions. Computational methods like molecular dynamics simulations can help create plausible unbound states for training and evaluation. Moreover, modeling protein flexibility and dynamics, which are critical in interactions, remains a challenge. Current methods are confined to rigid docking, but the dynamic nature of proteins in vivo means that we must account for flexibility to truly capture the essence of protein interactions.

On the computational side, there's a wealth of opportunities. Integrating evolutionary information from protein sequences could enhance the accuracy of our models. Combining sequence and structure data holds great promise, and attention mechanisms could help models identify relevant regions to focus on, enhancing both accuracy and interpretability. Exploring alternate loss functions, training strategies tailored for proteins, and architectural optimizations can further refine our models. Transfer learning from related data, like small molecule interactions, may also prove fruitful in enhancing the robustness of our predictions.

Functionally, there are numerous applications waiting to be explored. From epitope mapping and function prediction to mutation analysis and drug binding, the potential applications of our methods are vast. Challenges like membrane proteins and protein-nucleic acid interactions, which have historically been difficult to tackle, could be addressed with the methodologies developed in this thesis. Multimodal models that incorporate data from techniques like spectroscopy could provide even richer structural insights.

Perhaps the most exciting avenue lies in the realm of generative frameworks. Advances in protein language modeling open possibilities for conditional protein design and optimization. Techniques like reinforcement learning could be harnessed to discover completely new structural arrangements and interaction paradigms. The potential to not just understand, but also design and optimize proteins, the fundamental engines of life, is tantalizing.

In conclusion, the interface of biology and artificial intelligence offers a treasure trove of opportunities for discovery. This thesis has laid down foundational techniques to harness the patterns in protein structures for interaction prediction. By building upon these foundations, we can aspire to not just understand, but also design and innovate, pushing the boundaries of what we consider possible in protein science.

Bibliography

[1] Martın Abadi et al. “Tensorflow: a system for large-scale machine learning.” In: *Osdi*. Vol. 16. 2016. Savannah, GA, USA. 2016, pp. 265–283.

[2] Ruedi Aebersold and Matthias Mann. “Mass spectrometry-based proteomics”. In: *Nature* 422.6928 (2003), pp. 198–207.

[3] Ethan C Alley et al. “Unified rational protein engineering with sequence-based deep representation learning”. In: *Nature Methods* 16.12 (2019), pp. 1315–1322.

[4] Mohammed AlQuraishi. “End-to-end differentiable learning of protein structure”. In: *bioRxiv* (2018), p. 265231.

[5] Mohammed AlQuraishi. “End-to-end differentiable learning of protein structure”. In: *Cell Systems* 8.4 (2019), pp. 292–301.

[6] Stephen F Altschul et al. “Basic local alignment search tool”. In: *Journal of molecular biology* 215.3 (1990), pp. 403–410.

[7] Stephen F Altschul et al. “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs”. In: *Nucleic acids research* 25.17 (1997), pp. 3389–3402.

[8] Christian B Anfinsen. “Principles that govern the folding of protein chains”. In: *Science* 181.4096 (1973), pp. 223–230.

[9] Christof Angermueller et al. “Deep learning for computational biology”. In: *Molecular systems biology* 12.7 (2016), p. 878.

[10] Matan Atzmon, Haggai Maron, and Yaron Lipman. “Point convolutional neural networks by extension operators”. In: *arXiv:1803.10091* (2018).

[11] Žiga Avsec et al. “Effective gene expression prediction from sequence by integrating long-range interactions”. In: *Nature methods* 18.10 (2021), pp. 1196–1203.

[12] Utkarsh Ayachit. *The ParaView guide: a parallel visualization application*. Kitware, Inc., 2015.

[13] Tania A Baker et al. *Molecular biology of the gene*. Benjamin-Cummings Publishing Company, 2003.

[14] Jacob B Bale et al. “Accurate design of megadalton-scale two-component icosahedral protein complexes”. In: *Science* 353.6297 (2016), pp. 389–394.

[15] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. "Network medicine: a network-based approach to human disease". In: *Nature reviews genetics* 12.1 (2011), pp. 56–68.

[16] Alper Baspinar et al. "PRISM: a web server and repository for prediction of protein–protein interactions and modeling their 3D complexes". In: *Nucleic acids research* 42.W1 (2014), W285–W289.

[17] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv:1806.01261* (2018).

[18] Brian J Bender et al. "A practical guide to large-scale docking". In: *Nature protocols* 16.10 (2021), pp. 4799–4832.

[19] Helen Berman, Kim Henrick, and Haruki Nakamura. "Announcing the worldwide protein data bank". In: *Nature Structural & Molecular Biology* 10.12 (2003), pp. 980–980.

[20] Helen M Berman et al. "The protein data bank". In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[21] Surojit Biswas et al. "Low-N protein engineering with data-efficient deep learning". In: *bioRxiv* (2020).

[22] James F Blinn. "A generalization of algebraic surface drawing". In: *ACM TOG* 1.3 (1982), pp. 235–256.

[23] Joel R Bock and David A Gough. "Predicting protein–protein interactions from primary structure". In: *Bioinformatics* 17.5 (2001), pp. 455–460.

[24] Jaume Bonet Martınez et al. "Exploiting protein fragments in protein modelling and function prediction". PhD thesis. Universitat Pompeu Fabra, 2015.

[25] Alexandre MJJ Bonvin. "Flexible protein–protein docking". In: *Current opinion in structural biology* 16.2 (2006), pp. 194–200.

[26] Davide Boscaini et al. "Anisotropic diffusion descriptors". In: *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. 2016, pp. 431–441.

[27] Davide Boscaini et al. "Learning shape correspondence with anisotropic convolutional neural networks". In: *Proc. NIPS*. 2016.

[28] Michael M Bronstein et al. "Geometric deep learning: going beyond euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[29] Michael M Bronstein et al. "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges". In: *arXiv preprint arXiv:2104.13478* (2021).

[30] Yueqi Cao et al. "Efficient Curvature Estimation for Oriented Point Clouds". In: *arXiv:1905.10725* (2019).

[31] Benjamin Charlier et al. "Kernel operations on the GPU, with autodiff, without memory overflows". In: *arXiv:2004.11127* (2020).

[32] Jieming Chen, Nicholas Sawyer, and Lynne Regan. "Protein–protein interactions: General trends in the relationship between binding affinity and interfacial buried surface area". In: *Protein Science* 22.4 (2013), pp. 510–515.

[33] Rong Chen and Zhiping Weng. "Docking unbound proteins using shape complementarity, desolvation, and electrostatics". In: *Proteins: Structure, Function, and Bioinformatics* 47.3 (2002), pp. 281–294.

[34] Julian Chibane, Gerard Pons-Moll, et al. "Neural unsigned distance fields for implicit function learning". In: *Proc. NeurIPS*. 2020.

[35] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 539–546.

[36] Victor Chubukov et al. "Coordination of microbial metabolism". In: *Nature Reviews Microbiology* 12.5 (2014), pp. 327–340.

[37] Stephen R Comeau et al. "ClusPro: a fully automated algorithm for protein–protein docking". In: *Nucleic acids research* 32.suppl_2 (2004), W96–W99.

[38] Qian Cong et al. "Protein interaction networks revealed by proteome coevolution". In: *Science* 365.6449 (2019), pp. 185–189.

[39] Bruno E Correia et al. "Proof of principle for epitope-focused vaccine design". In: *Nature* 507.7491 (2014), p. 201.

[40] Gabriele Corso et al. "Diffdock: Diffusion steps, twists, and turns for molecular docking". In: *arXiv preprint arXiv:2210.01776* (2022).

[41] Sebastian Daberdaku and Carlo Ferrari. "Antibody interface prediction with 3D Zernike descriptors and SVM". In: *Bioinformatics* 35.11 (2019), pp. 1870–1876.

[42] Rhiju Das and David Baker. "Macromolecular modeling with rosetta". In: *Annu. Rev. Biochem.* 77 (2008), pp. 363–382.

[43] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56.

[44] David De Juan, Florencio Pazos, and Alfonso Valencia. "Emerging methods in protein co-evolution". In: *Nature Reviews Genetics* 14.4 (2013), nrg3414.

[45] Congyue Deng et al. "Vector neurons: A general framework for so (3)-equivariant networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12200–12209.

[46] Bruce R Donald. *Algorithms in structural molecular biology*. MIT Press, 2011.

[47] Tom Duff et al. "Building an orthonormal basis, revisited". In: *JCGT* 6.1 (2017).

[48] Dina Duhovny, Ruth Nussinov, and Haim J Wolfson. "Efficient unbound docking of rigid molecules". In: *Algorithms in Bioinformatics: Second International Workshop, WABI 2002 Rome, Italy, September 17–21, 2002 Proceedings 2*. Springer. 2002, pp. 185–200.

[49] James Dunbar et al. “SAbDab: the structural antibody database”. In: *Nucleic acids research* 42.D1 (2014), pp. D1140–D1146.

[50] Christiane Ehrt, Tobias Brinkjost, and Oliver Koch. “A benchmark driven guide to binding site comparison: An exhaustive evaluation using tailor-made data sets (ProSPECCTs)”. In: *PLoS computational biology* 14.11 (2018).

[51] Gökcen Eraslan et al. “Deep learning: new computational modelling techniques for genomics”. In: *Nature Reviews Genetics* 20.7 (2019), pp. 389–403.

[52] Matthias Fey and Jan E. Lenssen. “Fast Graph Representation Learning with PyTorch Geometric”. In: *Proc. ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019.

[53] Matthias Fey et al. “Splinecnn: Fast geometric deep learning with continuous b-spline kernels”. In: *Proc. CVPR*. 2018.

[54] Jean Feydy et al. “Fast geometric learning with symbolic matrices”. In: *Proc. NeurIPS* (2020).

[55] Martin J Field. “Simulating enzyme reactions: challenges and perspectives”. In: *Journal of computational chemistry* 23.1 (2002), pp. 48–58.

[56] Alexei V Finkelstein and Oleg Ptitsyn. *Protein physics: a course of lectures*. Elsevier, 2016.

[57] Sarel J Fleishman et al. “Computational design of proteins targeting the conserved stem region of influenza hemagglutinin”. In: *Science* 332.6031 (2011), pp. 816–821.

[58] Henry A Gabb, Richard M Jackson, and Michael JE Sternberg. “Modelling protein docking using shape complementarity, electrostatics and biochemical information”. In: *Journal of molecular biology* 272.1 (1997), pp. 106–120.

[59] P Gainza et al. “Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning”. In: *Nature Methods* (2019), pp. 1–9.

[60] Pablo Gainza et al. “De novo design of protein interactions with learned surface fingerprints”. In: *Nature* (2023), pp. 1–9.

[61] Pablo Gainza et al. “Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning”. In: *Nature Methods* 17.2 (2020), pp. 184–192.

[62] Octavian-Eugen Ganea et al. “Independent SE (3)-Equivariant Models for End-to-End Rigid Protein Docking”. In: *arXiv preprint arXiv:2111.07786* (2021).

[63] Wenhao Gao et al. “Deep Learning in Protein Structural Modeling and Design”. In: *arXiv:2007.08383* (2020).

[64] J. I. Garzon et al. “FRODOCK: a new approach for fast rotational protein-protein docking”. In: *Bioinformatics* 25.19 (Oct. 2009), pp. 2544–2551.

[65] Thomas Gaudelet et al. “Utilizing graph machine learning within drug discovery and development”. In: *Briefings in bioinformatics* 22.6 (2021), bbab159.

[66] George H Gauss et al. “The crystal structure of six-transmembrane epithelial antigen of the prostate 4 (Steap4), a ferri/cuprireductase, suggests a novel interdomain flavin-binding site”. In: *Journal of Biological Chemistry* 288.28 (2013), pp. 20668–20682.

[67] Mario Geiger and Tess Smidt. “e3nn: Euclidean neural networks”. In: *arXiv preprint arXiv:2207.09453* (2022).

[68] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

[69] Yulan Guo et al. “Deep learning for 3D point clouds: A survey”. In: *Trans. PAMI* (2020).

[70] Jun Yong Ha et al. “Crystal structure of D-erythronate-4-phosphate dehydrogenase complexed with NAD”. In: *Journal of molecular biology* 366.4 (2007), pp. 1294–1304.

[71] Pim de Haan et al. “Gauge Equivariant Mesh CNNs: Anisotropic convolutions on geometric graphs”. In: *arXiv:2003.05425* (2020).

[72] Mark A Hallen et al. “Osprey 3.0: open-source protein redesign for you, with powerful new features”. In: *Journal of computational chemistry* 39.30 (2018), pp. 2494–2507.

[73] Johannes C Hermann et al. “Structure-based activity prediction for an enzyme of unknown function”. In: *Nature* 448.7155 (2007), pp. 775–779.

[74] Lun Hu et al. “A survey on computational models for predicting protein–protein interactions”. In: *Briefings in bioinformatics* 22.5 (2021), bbab036.

[75] Po-Ssu Huang, Scott E Boyken, and David Baker. “The coming of age of de novo protein design”. In: *Nature* 537.7620 (2016), p. 320.

[76] John Ingraham et al. “Generative models for graph-based protein design”. In: *Proc. NeurIPS*. 2019.

[77] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *International Conference on Machine Learning*. 2015, pp. 448–456.

[78] Arian R Jamasb et al. “Graphein-a Python library for geometric deep learning and network analysis on protein structures and interaction networks”. In: *bioRxiv* (2020), pp. 2020–07.

[79] Lin Jiang et al. “De novo computational design of retro-aldol enzymes”. In: *science* 319.5868 (2008), pp. 1387–1391.

[80] Susan Jones and Janet M Thornton. “Prediction of protein-protein interaction sites using patch analysis”. In: *Journal of molecular biology* 272.1 (1997), pp. 133–143.

[81] Susan Jones and Janet M Thornton. “Principles of protein-protein interactions”. In: *Proceedings of the National Academy of Sciences* 93.1 (1996), pp. 13–20.

[82] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589.

[83] Elizabeth Jurrus et al. “Improvements to the APBS biomolecular solvation software suite”. In: *Protein Science* 27.1 (2018), pp. 112–128.

[84] Ephraim Katchalski-Katzir et al. “Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques”. In: *Proceedings of the National Academy of Sciences* 89.6 (1992), pp. 2195–2199.

[85] Lawrence A Kelley et al. “The Phyre2 web portal for protein modeling, prediction and analysis”. In: *Nature protocols* 10.6 (2015), pp. 845–858.

[86] Michel van Kempen et al. “Fast and accurate protein structure search with Foldseek”. In: *Nature Biotechnology* (2023), pp. 1–4.

[87] Mohamed Amine Ketata et al. “DiffDock-PP: Rigid Protein-Protein Docking with Diffusion Models”. In: *arXiv preprint arXiv:2304.03889* (2023).

[88] Daisuke Kihara et al. “Molecular surface representation using 3D Zernike descriptors for protein shape comparison and docking”. In: *Current Protein and Peptide Science* 12.6 (2011), pp. 520–530.

[89] Neil P King et al. “Computational design of self-assembling protein nanomaterials with atomic level accuracy”. In: *Science* 336.6085 (2012), pp. 1171–1174.

[90] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[91] Jan J Koenderink and Andrea J Van Doorn. “Surface shape and curvature scales”. In: *Image and vision computing* 10.8 (1992), pp. 557–564.

[92] Janez Konc et al. *ProBiS-CHARMMing: web interface for prediction and optimization of ligands in protein binding sites*. 2015.

[93] Tanja Kortemme, Alexandre V Morozov, and David Baker. “An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein–protein complexes”. In: *Journal of molecular biology* 326.4 (2003), pp. 1239–1259.

[94] Tanja Kortemme et al. “Computational redesign of protein-protein interaction specificity”. In: *Nature structural & molecular biology* 11.4 (2004), pp. 371–379.

[95] Zuzanna Kozicka and Nicolas Holger Thomä. “Haven’t got a glue: protein surface variation for the design of molecular glue degraders”. In: *Cell chemical biology* 28.7 (2021), pp. 1032–1047.

[96] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[97] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[98] John Kuriyan, Boyana Konforti, and David Wemmer. *The molecules of life: Physical and chemical principles*. Garland Science, 2012.

[99] Jack Kyte and Russell F Doolittle. “A simple method for displaying the hydropathic character of a protein”. In: *Journal of molecular biology* 157.1 (1982), pp. 105–132.

[100] Guillaume Launay and Thomas Simonson. “Homology modelling of protein-protein complexes: a simple method and its possibilities and limitations”. In: *BMC bioinformatics* 9.1 (2008), pp. 1–16.

[101] Michael C Lawrence and Peter M Colman. *Shape complementarity at protein/protein interfaces*. 1993.

[102] Adam Leach et al. “Denoising diffusion probabilistic models on so (3) for rotational alignment”. In: *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*. 2022.

[103] Andrew Leaver-Fay et al. “ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules”. In: *Methods in enzymology*. Vol. 487. Elsevier, 2011, pp. 545–574.

[104] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), p. 436.

[105] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. “Self-attention graph pooling”. In: *International conference on machine learning*. PMLR. 2019, pp. 3734–3743.

[106] Marc F Lensink, Sameer Velankar, and Shoshana J Wodak. “Modeling protein–protein and protein–peptide complexes: CAPRI 6th edition”. In: *Proteins: Structure, Function, and Bioinformatics* 85.3 (2017), pp. 359–377.

[107] Arthur Lesk. *Introduction to protein science: architecture, function, and genomics*. Oxford university press, 2010.

[108] Cyrus Levinthal. “Are there pathways for protein folding?” In: *Journal de chimie physique* 65 (1968), pp. 44–45.

[109] Yangyan Li et al. “PointCNN: Convolution on X-transformed points”. In: *Proc. NeurIPS*. 2018.

[110] Zeming Lin et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (2023), pp. 1123–1130.

[111] Zhihai Liu et al. “PDB-wide collection of binding data: current status of the PDBbind database”. In: *Bioinformatics* 31.3 (2015), pp. 405–412.

[112] Vivien Marx. “Method of the Year: protein structure prediction”. In: *Nature methods* 19.1 (2022), pp. 5–10.

[113] Jonathan Masci et al. “Geodesic convolutional neural networks on riemannian manifolds”. In: *Proceedings of the IEEE international conference on computer vision workshops*. 2015, pp. 37–45.

[114] Jonathan Masci et al. “Geometric deep learning”. In: *SIGGRAPH ASIA 2016 Courses*. ACM. 2016, p. 1.

[115] Simone Melzi et al. “GFrames: Gradient-based local reference frame for 3D shape matching”. In: *Proc. CVPR*. 2019.

[116] Francesco Milano et al. “Primal-Dual Mesh Convolutional Neural Networks”. In: *Proc. NeurIPS*. 2020.

[117] Federico Monti et al. “Fake news detection on social media using geometric deep learning”. In: *arXiv preprint arXiv:1902.06673* (2019).

[118] Federico Monti et al. “Geometric deep learning on graphs and manifolds using mixture model CNNs”. In: *Proc. CVPR*. Vol. 1. 2. 2017, p. 3.

[119] Garrett M Morris et al. “AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility”. In: *Journal of computational chemistry* 30.16 (2009), pp. 2785–2791.

[120] Marius Muja and David G Lowe. “Scalable nearest neighbor algorithms for high dimensional data”. In: *IEEE transactions on pattern analysis and machine intelligence* 36.11 (2014), pp. 2227–2240.

[121] Yoichi Murakami and Kenji Mizuguchi. “Applying the Naıve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites”. In: *Bioinformatics* 26.15 (2010), pp. 1841–1848.

[122] Denis Noble. “The rise of computational biology”. In: *Nature Reviews Molecular Cell Biology* 3.6 (2002), pp. 459–463.

[123] Thomas C Northey, Anja Barešić, and Andrew CR Martin. “IntPred: a structure-based predictor of protein–protein interaction sites”. In: *Bioinformatics* 34.2 (2018), pp. 223–229.

[124] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Proc. NeurIPS*. 2019.

[125] Eric F Pettersen et al. “UCSF Chimera—a visualization system for exploratory research and analysis”. In: *Journal of computational chemistry* 25.13 (2004), pp. 1605–1612.

[126] Brian Pierce and Zhiping Weng. “A combination of rescoring and refinement significantly improves protein docking performance”. In: *Proteins: Structure, Function, and Bioinformatics* 72.1 (2008), pp. 270–279.

[127] Brian Pierce and Zhiping Weng. “ZRANK: reranking protein docking predictions with an optimized energy function”. In: *Proteins: Structure, Function, and Bioinformatics* 67.4 (2007), pp. 1078–1086.

[128] Brian G Pierce, Yuichiro Hourai, and Zhiping Weng. “Accelerating protein docking in ZDOCK using an advanced 3D convolution library”. In: *PloS one* 6.9 (2011), e24657.

[129] Joan Planas-Iglesias et al. “Understanding protein–protein interactions using local structural features”. In: *Journal of molecular biology* 425.7 (2013), pp. 1210–1224.

[130] Aleksey Porollo and Jarosław Meller. "Prediction-based fingerprints of protein–protein interactions". In: *Proteins: Structure, Function, and Bioinformatics* 66.3 (2007), pp. 630–645.

[131] Adrien Poulenard and Maks Ovsjanikov. "Multi-directional geodesic neural networks via equivariant convolution". In: *ACM TOG* 37.6 (2018), pp. 1–14.

[132] Charles R Qi. *Deep learning on 3D data*. Springer, 2020.

[133] Charles R Qi et al. "PointNet: Deep learning on point sets for 3D classification and segmentation". In: *Proc. CVPR*. 2017.

[134] Charles Ruizhongtai Qi et al. "PointNet++: Deep hierarchical feature learning on point sets in a metric space". In: *Proc. NIPS*. 2017.

[135] Daniel Quang, Yifei Chen, and Xiaohui Xie. "DANN: a deep learning approach for annotating the pathogenicity of genetic variants". In: *Bioinformatics* 31.5 (2014), pp. 761–763.

[136] Frederic M Richards. "Areas, volumes, packing, and protein structure". In: *Annual review of biophysics and bioengineering* 6.1 (1977), pp. 151–176.

[137] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. "Octnet: Learning deep 3D representations at high resolutions". In: *Proc. CVPR*. 2017.

[138] David W Ritchie. "Recent progress and future directions in protein-protein docking". In: *Current protein and peptide science* 9.1 (2008), pp. 1–15.

[139] Tina Ritschel, Tom JJ Schirris, and Frans GM Russel. "KRIPO–a structure-based pharmacophores approach explains polypharmacological effects". In: *Journal of cheminformatics* 6.1 (2014), pp. 1–1.

[140] Alexander Rives et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *bioRxiv* (2019).

[141] Carol A Rohl et al. "Protein structure prediction using Rosetta". In: *Methods in enzymology*. Vol. 383. Elsevier, 2004, pp. 66–93.

[142] Jean-François Rual et al. "Towards a proteome-scale map of the human protein–protein interaction network". In: *Nature* 437.7062 (2005), pp. 1173–1178.

[143] Michel F Sanner, Arthur J Olson, and Jean-Claude Spehner. "Reduced surface: an efficient way to compute molecular surfaces". In: *Biopolymers* 38.3 (1996), pp. 305–320.

[144] Neil Savage. "Breaking into the black box of artificial intelligence". In: *Nature* (2022).

[145] Sawayamr. *Shapes of the 20 natural amino acids as they appear in an experimental electron density map at 1.5 angstrom resolution. Some amino acids have similar shapes, like threonine and valine, asparagine and aspartate, glutamine and glutamate*. 2002. URL: https://commons.wikimedia.org/wiki/File:20-amino-acids-density-map.jpg.

[146] Tamar Schlick. *Molecular modeling and simulation: an interdisciplinary guide: an interdisciplinary guide*. Vol. 21. Springer Science & Business Media, 2010.

[147] Gideon Schreiber and Sarel J Fleishman. "Computational design of protein–protein interactions". In: *Current opinion in structural biology* 23.6 (2013), pp. 903–910.

[148] LLC Schrodinger. "The PyMOL molecular graphics system". In: *Version* 1 (2015), p. 8.

[149] Andrew W Senior et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710.

[150] Thomas Shafee. *Hydrogen bonds in protein secondary structure. Cartoon above, atoms below with nitrogen in blue, oxygen in red (PDB: 1AXC)*. 2017. URL: https://commons.wikimedia.org/wiki/File:Alpha_beta_structure_(full).png.

[151] Kim A Sharp and Barry Honig. "Electrostatic interactions in macromolecules: theory and applications". In: *Annual review of biophysics and biophysical chemistry* 19.1 (1990), pp. 301–332.

[152] Alexandra Shulman-Peleg, Ruth Nussinov, and Haim J Wolfson. "Recognition of functional sites in protein structures". In: *Journal of molecular biology* 339.3 (2004), pp. 607–633.

[153] Song et al. "3D ShapeNets: A deep representation for volumetric shapes". In: *Proc. CVPR*. 2015.

[154] Hannes Stärk et al. "Equibind: Geometric deep learning for drug binding structure prediction". In: *International conference on machine learning*. PMLR. 2022, pp. 20503–20521.

[155] Freyr Sverrisson et al. "Fast end-to-end learning on protein surfaces". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15272–15281.

[156] Damian Szklarczyk et al. "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets". In: *Nucleic acids research* 47.D1 (2019), pp. D607–D613.

[157] Koichiro Tamura, Glen Stecher, and Sudhir Kumar. "MEGA11: molecular evolutionary genetics analysis version 11". In: *Molecular biology and evolution* 38.7 (2021), pp. 3022–3027.

[158] Maxim Tatarchenko et al. "Tangent convolutions for dense prediction in 3D". In: *Proc. CVPR*. 2018.

[159] Hugues Thomas et al. "KPconv: Flexible and deformable convolution for point clouds". In: *Proc. CVPR*. 2019.

[160] Julie D Thompson, Toby J Gibson, and Des G Higgins. "Multiple sequence alignment using ClustalW and ClustalX". In: *Current protocols in bioinformatics* 1 (2003), pp. 2–3.

[161] Raphael Townshend et al. "End-to-end learning on 3d protein structure for interface prediction". In: *Advances in Neural Information Processing Systems* 32 (2019).

[162] "UniProt: the universal protein knowledgebase". In: *Nucleic acids research* 45.D1 (2017), pp. D158–D169.

[163] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[164] Vishwesh Venkatraman et al. "Protein-protein docking using region-based 3D Zernike descriptors". In: *BMC bioinformatics* 10 (2009), pp. 1–21.

[165] Nitika Verma, Edmond Boyer, and Jakob Verbeek. "Feastnet: Feature-steered graph convolutions for 3d shape analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2598–2606.

[166] Thom Vreven et al. "Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2". In: *Journal of molecular biology* 427.19 (2015), pp. 3031–3041.

[167] Peng-Shuai Wang et al. "O-CNN: Octree-based convolutional neural networks for 3D shape analysis". In: *ACM TOG* 36.4 (2017), pp. 1–11.

[168] Yue Wang et al. "Dynamic graph cnn for learning on point clouds". In: *ACM TOG* 38.5 (2019), pp. 1–12.

[169] Joseph L Watson et al. "De novo design of protein structure and function with RFdiffusion". In: *Nature* (2023), pp. 1–3.

[170] Lingyu Wei et al. "Dense human body correspondences using convolutional networks". In: *Proc. CVPR*. 2016.

[171] J Michael Word et al. "Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation". In: *Journal of molecular biology* 285.4 (1999), pp. 1735–1747.

[172] Shiwen Wu et al. "Graph neural networks in recommender systems: a survey". In: *ACM Computing Surveys* 55.5 (2022), pp. 1–37.

[173] Wenxuan Wu, Zhongang Qi, and Li Fuxin. "PointConv: Deep convolutional networks on 3D point clouds". In: *Proc. CVPR*. 2019.

[174] Jinbo Xu. "Distance-based protein folding powered by deep learning". In: *PNAS* 116.34 (2019), pp. 16856–16865.

[175] Jianyi Yang et al. "Improved protein structure prediction using predicted interresidue orientations". In: *PNAS* 117.3 (2020), pp. 1496–1503.

[176] Jianyi Yang et al. "The I-TASSER Suite: protein structure and function prediction". In: *Nature methods* 12.1 (2015), pp. 7–8.

[177] Hai-Cheng Yi et al. "Graph representation learning in bioinformatics: trends, methods and applications". In: *Briefings in Bioinformatics* 23.1 (2022), bbab340.

[178] Shuangye Yin et al. "Fast screening of protein surfaces using geometric invariant fingerprints". In: *Proceedings of the National Academy of Sciences* (2009), pnas–0906146106.

[179] Manzil Zaheer et al. "Deep sets". In: *Proc. NIPS*. 2017.

[180] Qiangfeng Cliff Zhang et al. "Structure-based prediction of protein–protein interactions on a genome-wide scale". In: *Nature* 490.7421 (2012), pp. 556–560.

[181] Zuobai Zhang et al. “Protein representation learning by geometric structure pretraining”. In: *arXiv preprint arXiv:2203.06125* (2022).

[182] Jie Zhou et al. “Graph neural networks: A review of methods and applications”. In: *AI open* 1 (2020), pp. 57–81.

[183] Xiaolei Zhu, Yi Xiong, and Daisuke Kihara. “Large-scale binding ligand prediction by improved patch-based method Patch-Surfer2. 0”. In: *Bioinformatics* 31.5 (2015), pp. 707–713.

FREYR SVERRISSON

+41 76 455 80 79 ◇ freyrsverris@gmail.com
Rue Mercerie 4 ◇ 1003 Lausanne

EDUCATION

École Polytechnique Fédérale de Lausanne (EPFL) *March 2019 - Present*
Computational and Quantitative Biology, PhD.
Co-advised by Prof. Bruno Correia (EPFL) and Prof. Michael Bronstein (Imperial College London)
Swiss Data Science Center (SDSC) Fellowship
Visiting student at Imperial College London (*February 2020 - September 2020).*

École Polytechnique Fédérale de Lausanne (EPFL) September 2016 - October 2018
Bioengineering, M.Sc.
GPA 5.4/6
During my studies I did a number of projects in Prof. Bruno E. Correia's laboratory of protein design and immunoengineering. My initial projects got me acquainted with the core features and scripting interfaces of the Rosetta software suite, as well as some of the more advanced features such as multi-state design. My later projects included the design of scaffold based immunogens using the FunFolDes framework, developed in the lab. Later on I worked on a project for doing geometric deep learning on protein surfaces, which then became my master project. Our proposition is that the underlying fold of the protein is not necessarily the factor that determines the function of the protein, but rather the molecular surface it presents to its surroundings. In my masters project I looked at two applications of these methods in the field of protein design and analysis.

University of Iceland 2012-2015
Engineering Physics, B.Sc.
I graduated in engineering physics in spring 2015 (GPA 8.9/10) and in the following academic year I took additional courses in molecular biology. I did my final project under the supervision of Prof. Steinn Gudmundsson. The project was in metabolic systems biology and had the title Estimation of metabolite concentrations in metabolic networks with Monte Carlo simulation (the thesis is in Icelandic). In the project I made an attempt to estimate metabolite concentrations in a model of the central metabolism of E. coli by random sampling of thermodynamically feasible states.

Reykjavik Junior College 2008-2012
I graduated from the physics department.

EXPERIENCE

Isomorphic Labs Research Intern *June 2022 - November 2022*

EPFL Research Intern *October 2018 - March 2019*
Since I graduated I worked in Prof. Bruno E. Correia's laboratory to further develop the methods I worked on for my master's thesis and to explore other applications of them.

Internship in Alvotech (2 months) *2018*
During my masters I did a two months internship in Alvotech, a biopharmaceutical company. My project there was to develop CD and FT-IR spectra methods and standard operating procedures (SOPs) for determining biosimilarity. I also got training and experience in working in a GMP facility. I developed protocols both for collecting high quality spectra and for comparing denaturation pathways.

Participation in Startup Reykjavik Accelerator Program *2015*
I started a company with some friends and we got into the Startup Reykjavik accelerator (10 chosen out of 150 applications). The idea was to build a web platform for people to make predictions about the future. The company is not running today but during the program we had lectures, met with mentors and learned much about how to start a company.

University of Iceland *2013-2015*
I worked as a teaching assistant in the following courses: Numerical Analysis (Spring 2015), Mathematical Analysis IC (Fall 2014, Fall 2013), Physics 2 V (Spring 2014) and Physics 1 V (Fall 2013).

Center of Systems Biology (University of Iceland) *Summer 2014*
I did an internship in Prof. Sigurdur Brynjolfsson laboratory at the university's Center for Systems Biology. I and another student had the project of building a prototype of a photobioreactor for growing algae. This photobioreactor was supposed to take emissions from geothermal energy plants as an input and fix the carbon from them. We designed and built electrical circuits and software of a control system interacting with mechanical valves and sensors. We developed a way of measuring algae density using phototransistors and light-emitting diodes. For the project we eventually we got the second place in University of Icelands Applied Science Price.

The Ingvarsson research group (University of Iceland) *Summer 2013*
The summer after my first year at university I spent three months in Prof. Snorri Ingvarsson laboratory as an intern. Prof. Ingvarsson is a professor of physics and was researching the electromagnetic and thermal properties of nanostructures. He would use sputtering for thin-film deposition and perform various measurements on them, mostly for the purpose of microsensing and magnetic memory storage. My responsibilities included performing these kind of measurements, for instance using x-ray diffraction for determining film composition and measuring changes in light polarisation for characterizing magnetic hysteresis. During this time I also got some experience in working in a clean room.

Nobel 101 (Icelandic: Nóbel námsbúdir) *2012-2013*
I taught physics courses (preparation for finals) for students in gymnasium (age 18-20).

TECHNICAL SKILLS

Programming Languages	Python, MATLAB, C/C++, Bash, Mathematica
Python Packages	Pandas, Matplotlib, Numpy, Scipy, Scikit-learn, Tensorflow, PyTorch, Seaborn, Dask, Numba, Jupyter
Software & Tools	Rosetta, LaTeX, Pymol, AMBER, MS Office, AutoCAD

PUBLICATIONS

Sverrisson, F., Feydy, J., Correia, B. E., Bronstein, M. M. (2021). Fast end-to-end learning on protein surfaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 15272-15281) (Best Paper Candidate).

Gainza, P, Sverrisson, F, Monti, F, Rodola, E, Boscaini, D, Bronstein, MM, Correia, BE (2020). Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nature Methods, 17(2), 184-192.

Gainza P, Sverrisson F, Monti F, Rodola E, Boscaini D, Bronstein MM, Correia BE. Learning interaction patterns from surface representations of protein structure. NeurIPS 2019, Workshop on Graph Representation Learning.

Bonet, Jaume, et al. "Rosetta FunFolDesA general framework for the computational design of functional proteins." PLoS computational biology 14.11 (2018): e1006623.

OTHER

Co-organizer of AI & Molecular World track at AMLD	*2019*
Recipient of the SDSC fellowship	*2019*
Master student talk and a poster at EPFL Bioengineering Day	*2017*
Second place in University of Iceland's Applied Science Prize	*2015*
Awarded the Icelandic Student Services Project Grant	*2014*
Participation in the International Physics Olympiad	*2012*