Multi-Scale Representation Learning on Proteins

Vignesh Ram Somnath*
Dept. of Computer Science
ETH Zurich
vsomnath@ethz.ch

Charlotte Bunne*
Dept. of Computer Science
ETH Zurich
bunnec@ethz.ch

Andreas Krause
Dept. of Computer Science
ETH Zurich
krausea@ethz.ch

Abstract

Proteins are fundamental biological entities mediating key roles in cellular function and disease. This paper introduces a multi-scale graph construction of a protein – HOLOPROT – connecting surface to structure and sequence. The surface captures coarser details of the protein, while sequence as primary component and structure – comprising secondary and tertiary components – capture finer details. Our graph encoder then learns a multi-scale representation by allowing each level to integrate the encoding from level(s) below with the graph at that level. We test the learned representation on different tasks, (i.) ligand binding affinity (*regression*), and (ii.) protein function prediction (*classification*). On the regression task, contrary to previous methods, our model performs consistently and reliably across different dataset splits, outperforming all baselines on most splits. On the classification task, it achieves a performance close to the top-performing model while using 10x fewer parameters. To improve the memory efficiency of our construction, we segment the multiplex protein surface manifold into *molecular* superpixels and substitute the surface with these superpixels at little to no performance loss.

1 Introduction

Protein design and engineering has become a crucial component of pharmaceutical research and development, finding application in a wide variety of diagnostic and industrial settings. Besides understanding the design principles determining structure and function of proteins, current efforts seek to further enhance or discover proteins with properties useful for technological or therapeutic applications. To efficiently guide the search in the vast design space of functional proteins, we need to be able to robustly predict properties of a candidate protein [Yang et al., 2019]. Moreover, understanding role and function of proteins is crucial to study causes and mechanism of human disease [Fessenden, 2017].

To achieve this, representations incorporating the complex nature of proteins are required. Proteins consist of amino acids, organic molecules linked by peptide bonds forming a linear *sequence*. Each of the twenty amino acids carries a unique side chain, giving rise to an incomprehensibly large combinatorial space of possible protein sequences. The primary sequence drives the folding of polymers – a spontaneous process guided by hydrophobic interactions, formation of intramolecular hydrogen bonds, and van der Waals forces into a unique three-dimensional *structure*. The resulting shape and *surface* manifold with rich physiochemical properties carry essential information for understanding function and potential molecular interactions.

Previous methods typically only consider an individual subset within these scales, focusing on either sequence [Öztürk et al., 2018, Hou et al., 2018], three-dimensional structure [Hermosilla et al., 2021, Derevyanko et al., 2018] or surface [Gainza et al., 2020]. Two proteins with similar sequences can fold into entirely different conformations. While these proteins might catalyze the same type of

*Equal contribution.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

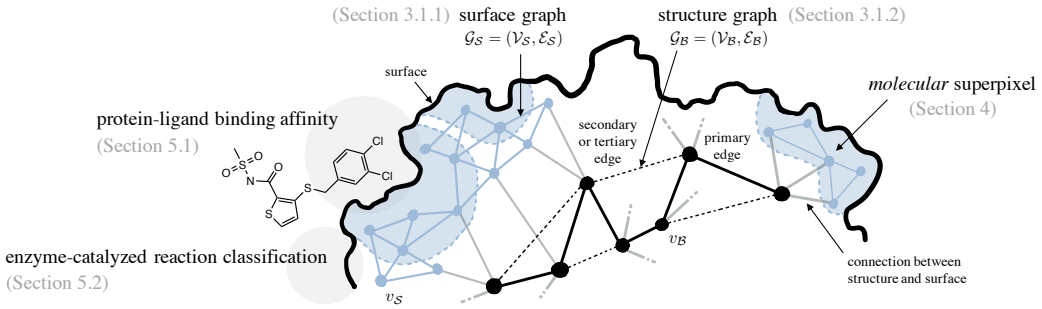

Figure 1: **Overview of HOLOPROT** Our multi-scale protein representation algorithm integrates primary, secondary and tertiary elements of protein structures and connects them to the surface. We extract higher-level protein motifs by introducing *molecular* superpixels. Both structure and surface are represented as graphs $\mathcal{G}_\mathcal{B}$ and $\mathcal{G}_\mathcal{S}$, respectively. The method is evaluated on two representative tasks, protein-ligand binding affinity and enzyme-catalyzed reaction classification.

reactions, their behavior to specific inhibiting drugs might be divergent. Interaction between proteins and ligands, on the other hand, is controlled by molecular surface contacts [Gainza et al., 2020]. Molecular surfaces, determined by subjacent amino acids, are fingerprinted with patterns of geometric and chemical properties, and thus their integration in protein representations is crucial.

In this work, we present a novel multi-scale graph representation which integrates and connects the complex nature of proteins across all levels of information. HOLOPROT consists of a surface and structure layer (both represented as graphs) with explicit edges between the layers. Our construction is guided by the intuition that propagating information from surface to structure would allow each residue to learn encodings reflective of not just its immediate residue neighborhood, but also the higher-level geometric and chemical properties that arise from interactions between a residue and its neighborhood. The associated multi-scale encoder then learns representations by integrating the encoding from the layer below, with the graph at that layer (Section 3). Such multi-scale representations have been previously used in molecular graph generation [Jin et al., 2020] with impressive results.

We further improve the memory efficiency of our construction by segmenting the large and rich protein surface into *molecular "superpixels"*, summarizing higher-level fingerprint features and motifs of proteins. Substituting the surface layer with these superpixels results in little to no performance degradation across the evaluated tasks. The concept of *molecular superpixels* might be of interest beyond our model (Section 4).

The multi-objective and multi-task nature of protein engineering poses a challenge for current methods, often designed and evaluated only on specific subtasks of protein design. By incorporating the biology of proteins, strong representations exhibit robust performance across tasks. We demonstrate our model's versatility and range of applications by deploying it to tasks of rather distinct nature, including a regression task, e.g., inference of protein ligand binding affinity, and classification tasks, i.e., enzyme-catalyzed reaction classification (Section 5).

2 Related Work

Protein Representation Learning With increasing availability of sequence and structure data, the field of protein representation learning has advanced rapidly, with methods falling largely in one of the following categories:

Sequence-based methods. One-dimensional amino acid sequences continue to be the simplest, most abundant source of protein data and various methods have been developed that borrow architectures developed in natural language processing (NLP). One-dimensional convolutional neural networks have been used to classify a protein sequence into folds and enzyme function [Hou et al., 2018, Dalkiran et al., 2018], and to predict their binding affinity to ligands [Öztürk et al., 2018]. Furthermore, methods have applied complex NLP models trained unsupervised on millions of unlabeled protein sequences and fine-tuned them on different downstream tasks [Rao

et al., 2019, Elnaggar et al., 2020, Bepler and Berger, 2019]. Despite being advantageous when only the sequence is available, these methods ignore the full spatial complexity of proteins.

Structure-based methods. To learn beyond sequences, approaches have been developed, that consider the 3D structure of proteins. 3D convolutional neural networks have been utilized for protein quality assessment [Derevyanko et al., 2018], protein contact prediction [Townshend et al., 2019] and protein-ligand binding affinity tasks [Ragoza et al., 2017, Jiménez et al., 2018, Townshend et al., 2020]. An alternate representation treats proteins as graphs, applying graph neural networks for enzyme classification [Dobson and Doig, 2005], interface prediction [Fout et al., 2017], and protein structure quality prediction [Baldassarre et al., 2021]. Gligorijevic et al. [2021] use a long short term memory cell (LSTM) to encode the sequence, followed by a graph convolutional network (GCN) [Kipf and Welling, 2017] to capture the tertiary structure, and apply this to the function prediction task. Hermosilla et al. [2021] propose a convolutional operator that learns to adapt filters based on the primary, secondary, and tertiary structure of a protein, showing strong performance on reaction and fold class prediction.

Surface-based methods. Taking a different viewpoint, Gainza et al. [2020] hypothesize that the protein surface displays patterns of chemical and geometric features that fingerprint a protein's interaction with other biomolecules. They utilize geodesic convolutions, which are extensions of convolutions on surfaces, and learn fingerprint vectors, showing improved performance across binding pocket and protein interface prediction tasks.

Protein Motif Detection Protein motifs have largely been synonymous with common and conserved patterns in a protein's sequence or structure influencing protein function, e.g., the helix-turn-helix motif binds DNA. Understanding these fragments is essential for 3D structure prediction, modeling, and drug design. While reliably detecting evolutionary motifs, existing tools [Golovin and Henrick, 2008] do not provide a full segmentation of the protein surface manifold. Our work takes a different viewpoint, by looking at protein motifs from the context of a protein surface. Previous methods developed in this context either only consider geometric information rather than physiological properties [Cantoni et al., 2010], are computationally expensive [Cantoni et al., 2011], or designed for particular downstream tasks [Stepniewska-Dziubinska et al., 2020]. Our molecular superpixel approach provides a task-independent segmentation utilizing both geometric and chemical features, while also being computationally efficient.

3 Multi-Scale Protein Representation

In this section, we describe our multi-scale graph construction and the associated encoder. Figure 1 illustrates the main principles of HOLOPROT. We represent a protein $\mathcal{P}$ as a graph $\mathcal{G}_\mathcal{P}$ with two layers capturing different scales:

(i.) **Surface layer.** This layer captures the coarser representation details of a protein. The protein surface is generated using the triangulation software MSMS [Connolly, 1983, Sanner et al., 1996]. We represent this layer as a graph $\mathcal{G}_\mathcal{S}$, where each surface node $u_\mathcal{S}$ has a feature vector $\mathbf{f}_{u_\mathcal{S}}$ denoting its charge, hydrophobicity and local curvature [Gainza et al., 2020]. Two surface nodes $(u_\mathcal{S}, v_\mathcal{S})$ have an edge if they are part of a triangulation. Each surface node additionally has a residue identifier r, indicating the amino acid residue it corresponds to. Multiple surface nodes can have the same residue identifier.

(ii.) **Structure layer.** This layer captures the finer representation details of a protein. A protein typically has four structural levels: (i.) primary structure (sequence), (ii.) secondary structure (α-helices and β-sheets), (iii.) tertiary structure (3D structure) and (iv.) quaternary structure (complexes) [Fout et al., 2017]. We represent this layer as a graph $\mathcal{G}_\mathcal{B}$, where each node $u_\mathcal{B}$ corresponds to a residue r. Two nodes $(u_\mathcal{B}, v_\mathcal{B})$ have an edge in $\mathcal{G}_\mathcal{B}$ if the C_α atoms of the two nodes occur within a certain distance of each other. Distance based thresholding ensures that different structural levels are implicitly captured in the neighborhood of a node $u_\mathcal{B}$.

We further introduce edges from the surface layer to the structure layer in order to propagate information between them. Specifically, we introduce a directed edge between a surface node $u_\mathcal{S}$ and a backbone node $u_\mathcal{B}$ if they both have the same residue identifier r. Typically, we have between 20-40 surface nodes $\{u_\mathcal{S}\}$ that map to the same structure node $u_\mathcal{B}$. This gives us the multi-scale

graph which is then encoded by our multi-scale message passing network. Details on the features used for both the structure and surface layer can be found in Appendix D.1.

3.1 Multi-Scale Encoder

Our multi-scale message passing network uses one *message passing neural network (MPN)* for each layer in the multi-scale graph [Lei et al., 2017, Gilmer et al., 2017]. This allows us to learn structured representations of each scale, which can then be tied together through connections between the scales. Before detailing the remainder of the architecture, we introduce some notational preliminaries. For simplicity, we denote the MPN encoding process as $\text{MPN}_\theta(\cdot)$ with parameters θ. We denote $\text{MLP}_\theta(\mathbf{x}, \mathbf{y})$ for a multi-layer perceptron (MLP) with parameters θ, whose input is the concatenation of $\mathbf{x}$ and $\mathbf{y}$, and $\text{MLP}_\theta(\mathbf{x})$ when the input is only $\mathbf{x}$. We also denote the residue identifier of a node u with $\text{id}(u)$, and the neighbors of a node u as $\mathcal{N}(u)$. The details of the MPN architecture are listed in the Appendix A.

3.1.1 Surface Message Passing Network

We first encode the surface layer $\mathcal{G}_\mathcal{S}$ of the multi-scale protein graph $\mathcal{G}_\mathcal{P}$. The inputs to the MPN are node features $\mathbf{f}_{u_\mathcal{S}}$ and edge features $\mathbf{f}_{u_\mathcal{S} v_\mathcal{S}}$ of $\mathcal{G}_\mathcal{S}$. For more details on the input features used for surface nodes and edges, refer to Appendix D.1. The MPN (with parameters $\theta_\mathcal{S}$) propagates messages between the nodes for K iterations, and outputs a representation $h_{u_\mathcal{S}}$ for each surface node $u_\mathcal{S}$,

$$\{\mathbf{h}_{u_\mathcal{S}}\} = \text{MPN}_{\theta_\mathcal{S}}(\mathcal{G}_\mathcal{S}, \{\mathbf{f}_{u_\mathcal{S}}\}, \{\mathbf{f}_{u_\mathcal{S} v_\mathcal{S}}\}_{v_\mathcal{S} \in \mathcal{N}(u_\mathcal{S})}).$$

3.1.2 Structure Message Passing Network

For each node $u_\mathcal{B}$ in the structure layer $\mathcal{G}_\mathcal{B}$, we first prepare the input to the MPN (with parameters $\theta_\mathcal{B}$) by using an MLP (with parameters θ) on the concatenated version of its initial features $\mathbf{f}_{u_\mathcal{B}}$ and the mean of the surface node vectors with the same residue identifier $S = \{\mathbf{h}_{u_\mathcal{S}} | \text{id}(u_\mathcal{S}) = \text{id}(u_\mathcal{B})\}$

$$\mathbf{x}_{u_\mathcal{B}} = \text{MLP}_\theta(\mathbf{f}_{u_\mathcal{B}}, \textstyle\sum_S \mathbf{h}_{u_\mathcal{S}} / |S|).$$

Given the edge features $\mathbf{f}_{u_\mathcal{B} v_\mathcal{B}}$, we then run K iterations of message passing, to compute the representations $\mathbf{h}_{u_\mathcal{B}}$ for each structure node $u_\mathcal{B}$,

$$\{\mathbf{h}_{u_\mathcal{B}}\} = \text{MPN}_{\theta_\mathcal{B}}(\mathcal{G}_\mathcal{B}, \{\mathbf{x}_{u_\mathcal{B}}\}, \{\mathbf{f}_{u_\mathcal{B} v_\mathcal{B}}\}_{v_\mathcal{B} \in \mathcal{N}(u_\mathcal{B})}).$$

The graph representation $\mathbf{c}_{\mathcal{G}_\mathcal{P}}$ is an aggregation of structure node representations,

$$\mathbf{c}_{\mathcal{G}_\mathcal{P}} = \sum_{u_\mathcal{B} \in \mathcal{G}_\mathcal{B}} \mathbf{h}_{u_\mathcal{B}}. \tag{1}$$

3.2 Task Specific Training

This multi-scale encoding allows us to learn a structured representation of a protein tying different scales together, which can then be utilized for any downstream task. In this work, we evaluate our method on two rather distinct tasks (i.) protein-ligand binding affinity regression, and (ii.) enzyme–catalyzed reaction classification. The architectural details for both downstream tasks are described below. These modules can be adapted and modified in order to utilize HOLOPROT for other use cases.

3.2.1 Protein-Ligand Binding Affinity

Protein-ligand binding affinity prediction depends on the interaction of a protein, encoded using the HOLOPROT framework, and a corresponding ligand, in most cases small molecules. To encode the ligand represented as a graph $\mathcal{G}_\mathcal{L}$, we use another MPN (with parameters $\theta_\mathcal{L}$) and aggregate its node representations to obtain a graph representation $c_{\mathcal{G}_\mathcal{L}}$. We concatenate the graph representations $\mathbf{c}_{\mathcal{G}_\mathcal{P}}$ (Equation 1) of the protein and $c_{\mathcal{G}_\mathcal{L}}$ of the ligand, and use that as input to a MLP (with parameters ϕ) to obtain predictions,

$$s_a = \text{MLP}_\phi(c_{\mathcal{G}_\mathcal{P}}, c_{\mathcal{G}_\mathcal{L}}). \tag{2}$$

The model is trained by minimizing the mean squared error.

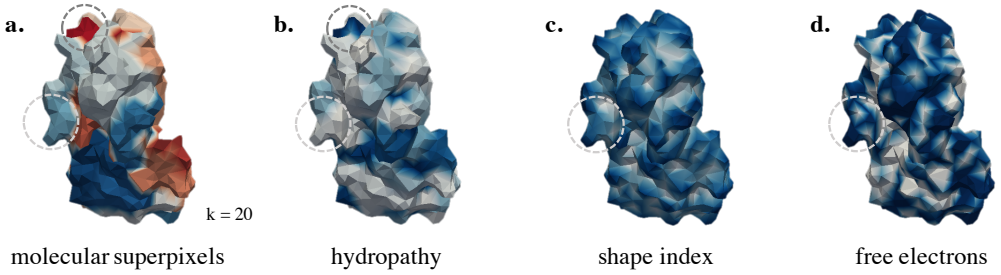

Figure 2: **Molecular Superpixels and Surface Features of the HIV-1 Protease** (`PDB ID: 2AVQ`)**.** **a.** Molecular superpixels, indicated by different colors ($k = 20$), and the corresponding surface features, i.e., **b.** hydropathy, **c.** shape index, and **d.** free electrons. As highlighted, molecular superpixels are spatially compact and overlap with surface regions dominated by single features such as hydrophobic patches while capturing coherent areas across all surface features. The protein complex contains 198 residues.

3.2.2 Enzyme-Catalyzed Reaction Classification

To predict the enzyme-catalyzed reaction class, we use the graph representation $\mathbf{c}_{\mathcal{G}_\mathcal{P}}$ of the protein obtained via HOLOPROT as the input to a MLP (with parameters ϕ) to obtain the prediction logits,

$$p_k = \text{MLP}_\phi(c_\mathcal{G}). \tag{3}$$

The model is trained by minimizing the cross-entropy loss.

4 Superpixels on Molecular Surfaces

Protein surface manifolds are complex and represented via large meshes. In order to improve the computational and memory efficiency of our construction, we introduce the notion of *molecular superpixels*. Originally developed in computer vision [Ren and Malik, 2003, Mori et al., 2004, Kohli et al., 2009], superpixels are defined as perceptually uniform regions in the image. In the molecular context, we refer to superpixels as segments on the protein surface capturing higher-level fingerprint features and protein motifs such as hydrophobic binding sites.

In order to apply the segmentation principle to three-dimensional molecular surfaces, we employ graph-based superpixel algorithms on triangulated surface meshes. The superpixel representation of the protein surface needs to satisfy several requirements, as (i.) molecular superpixels should not reduce the overall achievable performance of HOLOPROT, and (ii.) molecular superpixels need to form geometrically compact clusters, and overlap with surface regions that are coherent in physiological surface properties, e.g., capture hydrophobic binding sides or highly charged areas. Popular graph-based segmentation tools such as Felzenszwalb and Huttenlocher [2004, FH], mean shift [Comaniciu and Meer, 2002], and watershed [Vincent and Soille, 1991], however, produce non-compact superpixels of irregular sizes and shapes. By posing the segmentation task as a maximization problem on a graph maximizing over (i.) the entropy rate of the random walk on the surface graph $\mathcal{G}_\mathcal{S} = (\mathcal{V}_\mathcal{S}, \mathcal{E}_\mathcal{S})$ favoring the formation of compact and homogeneous clusters, and (ii.) a balancing term encouraging clusters with similar sizes, the entropy rate superpixel (ERS) segmentation algorithm [Liu et al., 2011] outperforms previous methods across different tasks [Stutz et al., 2018] and achieves the desired properties of *molecular* superpixels.

In order to incorporate geometric and chemical features of the surface $\mathbf{F}_\mathcal{S}$, we extend the surface graph $\mathcal{G}_\mathcal{S} = (\mathcal{V}_\mathcal{S}, \mathcal{E}_\mathcal{S})$ with a non-negative similarity measure w, given as $w_{ij} = \sum_{\mathbf{f}\in\mathbf{F}_\mathcal{S}} |\mathbf{f}_{v_i}\mathbf{f}_{v_j}|$ for nodes v_i and v_j if connected by an edge e_{ij}. We simulate a random walk $\mathbf{X} = \{X_t | t \in T, X_t \in \mathcal{V}_\mathcal{S}\}$ on a protein surface mesh, where the transition probability p_{ij} between two nodes v_i and v_j is defined as $p_{ij} = P(X_{t+1} = v_j | X_t = v_i) = {}^{w_{ij}}/_{w_i}$, where $w_i = \sum_{k:e_{ik}\in\mathcal{E}_\mathcal{S}} w_{ik}$.The corresponding stationary distributions of nodes $\mathcal{V}_\mathcal{S}$ are given by

$$\boldsymbol{\mu} = \left(\mu_1, \mu_2, \ldots, \mu_{|\mathcal{V}_\mathcal{S}|}\right)^\top = \left(\frac{w_1}{w_T}, \frac{w_2}{w_T}, \ldots, \frac{w_{|\mathcal{V}_\mathcal{S}|}}{w_T}\right)^\top.$$

Molecular superpixels are then defined by a subset of edges $\mathcal{M} \subseteq \mathcal{E}_\mathcal{S}$ such that the resulting graph, $\mathcal{G}_\mathcal{S} = (\mathcal{V}_\mathcal{S}, \mathcal{M})$, contains exactly k connected subgraphs. Computing *molecular* superpixels is achieved via optimizing the objective function with respect to the edge set $\mathcal{M}$

$$\max_{\mathcal{M}} \underbrace{-\sum_i \mu_i \sum_j p_{ij}(\mathcal{M}) \log\left(p_{ij}(\mathcal{M})\right)}_{\text{(i.) entropy rate}} \underbrace{-\sum_i p_{Z_\mathcal{M}}(i) \log\left(p_{Z_\mathcal{M}}(i)\right) - n_\mathcal{M}}_{\text{(ii.) balancing function}}$$

$$\text{s.t. } \mathcal{M} \subseteq \mathcal{E}_\mathcal{S} \text{ and } n_\mathcal{M} \geq k,$$

where $n_\mathcal{M}$ is the number of connected components in the graph, $p_{Z_\mathcal{M}}$ denotes the distribution of cluster memberships $Z_\mathcal{M}$, and $\lambda \geq 0$ is the weight of the balancing term. Both terms satisfy monotonicity and submodularity and can thus be efficiently optimized based on techniques from submodular optimization [Nemhauser et al., 1978]. For further details on the entropy rate superpixel algorithm, see Liu et al. [2011].

A molecular superpixel m comprising k surface vertices is then given as $\mathbf{f}_m = (\mathbf{f}_{v_1}, \ldots, \mathbf{f}_{v_k})$ for all $\mathbf{f} \in \mathbf{F}_\mathcal{S}$. We summarize the feature representation of each molecular superpixel via the graph $\mathcal{G}_\mathcal{M} = (\mathcal{V}_\mathcal{M}, \mathcal{E}_\mathcal{M})$, where each node $m \in \mathcal{V}_\mathcal{M}$ is represented via ($\texttt{mean}(\mathbf{f}_m)$, $\texttt{std}(\mathbf{f}_m)$, $\texttt{max}(\mathbf{f}_m)$, $\texttt{min}(\mathbf{f}_m)$) for all $\mathbf{f} \in \mathbf{F}_\mathcal{S}$ and an edge $e \in \mathcal{E}_\mathcal{M}$ via the Wasserstein distance between neighboring superpixels.

Figure 2 demonstrates *molecular* superpixels for the enzyme HIV-1 protease [Brik and Wong, 2003]. Besides being spatially compact, superpixels overlap with surface regions dominated by single features such as hydrophobic patches, while capturing coherent areas across all surface features. Further examples of superpixels are displayed in Appendix C.2.

5 Evaluation

Successful protein engineering requires optimization of multiple objectives. When searching for a protein with desired functionality, auxiliary but crucial properties such as stability measured in terms of free energy of folding also need to be satisfied. Furthermore, the field is also subject to a plethora of potential tasks and applications. In order to capture the multi-objective and multi-task nature of protein engineering, we evaluate our method on two representative tasks: regression of the binding affinity between proteins and their ligands, and classification of enzyme proteins based on the type of reaction they catalyze.

5.1 Protein-Ligand Binding Affinity Prediction

Studying the interaction between proteins and small molecules is crucial for many downstream tasks, e.g., accelerating virtual screening for potential candidates in drug discovery or protein design to improve the output of an enzyme-catalyzed reaction. The architecture of the regression module is described in Equation 2.

Dataset. The PDBBIND database (version 2019) [Liu et al., 2017] is a collection of the experimentally measured binding affinity data for all types of biomolecular complexes deposited in the Protein Data Bank [Berman et al., 2000]. After quality filtering for resolution and surface construction, the refined subset comprises a total of $4,709$ biomolecular complexes. The binding affinity provided in PDBBIND is experimentally determined and expressed in molar units of the inhibition constant (K_i) or dissociation constant (K_d). Similar to previous methods [Öztürk et al., 2018, Townshend et al., 2020], we do not distinguish both constants and predict negative log-transformed binding affinity pK_d/pK_i. We split the dataset into training, test and validation splits based on the scaffolds of the corresponding ligands (*scaffold*), or a 30% and a 60% sequence identity threshold (*identity 30%*, *identity 60%*) to limit homologous ligands or proteins appearing in both train and test sets.

Baselines. For evaluating the overall performance on the regression task, we compare HOLOPROT against several baselines including current state-of-the-art methods on both tasks. This comprises sequence-based methods [Öztürk et al., 2018, Rao et al., 2019, Bepler and Berger, 2019, Elnaggar et al., 2020] as well as methods based on the three-dimensional structure of proteins [Townshend et al., 2020, Hermosilla et al., 2021], and recent methods using geometric deep learning on protein molecular surfaces [Gainza et al., 2020].

Table 1: **Protein-Ligand Binding Affinity Prediction Results** Comparison predictive performance of ligand binding affinity using the PDBbind dataset [Liu et al., 2017] of HOLOPROT against other methods. Results are reported for 3 experimental runs.

Model	**# Params**	**Sequence Identity (30 %)**			**Sequence Identity (60 %)**		
		RMSE	Pearson	Spearman	RMSE	Pearson	Spearman
Sequence-based Methods							
Öztürk et al. [2018]	1.93 M	1.866 ± 0.080	0.472 ± 0.022	0.471 ± 0.024	1.762 ± 0.261	0.666 ± 0.012	0.663 ± 0.015
Bepler and Berger [2019]	48.8 M	1.985 ± 0.006	0.165 ± 0.006	0.152 ± 0.024	1.891 ± 0.004	0.249 ± 0.006	0.275 ± 0.008
Rao et al. [2019]	93.0 M	1.890 ± 0.035	0.338 ± 0.044	0.286 ± 0.124	1.633 ± 0.016	0.568 ± 0.033	0.571 ± 0.021
Elnaggar et al. [2020]	2.4M[1]	1.544 ± 0.015	0.438 ± 0.053	0.434 ± 0.058	1.641 ± 0.016	0.595 ± 0.014	0.588 ± 0.009
Surface-based Methods							
Gainza et al. [2020]	0.62 M	1.484 ± 0.018	0.467 ± 0.020	0.455 ± 0.014	1.426 ± 0.017	0.709 ± 0.008	0.701 ± 0.011
Structure-based Methods							
Townshend et al. [2020][2]	-	$\mathbf{1.429 \pm 0.042}$	0.541 ± 0.029	0.532 ± 0.033	1.450 ± 0.024	0.716 ± 0.008	0.714 ± 0.009
Townshend et al. [2020][3]	-	1.936 ± 0.120	$\mathbf{0.581 \pm 0.039}$	$\mathbf{0.647 \pm 0.071}$	1.493 ± 0.010	0.669 ± 0.013	0.691 ± 0.010
Hermosilla et al. [2021]	5.80 M	1.554 ± 0.016	0.414 ± 0.053	0.428 ± 0.032	1.473 ± 0.024	0.667 ± 0.011	0.675 ± 0.019
HOLOPROT (●)	1.44 M	1.464 ± 0.006	0.509 ± 0.002	0.500 ± 0.005	$\mathbf{1.365 \pm 0.038}$	$\mathbf{0.749 \pm 0.014}$	$\mathbf{0.742 \pm 0.011}$
HOLOPROT (◆)	1.76 M	1.491 ± 0.004	0.491 ± 0.014	0.482 ± 0.017	1.416 ± 0.022	0.724 ± 0.011	0.715 ± 0.006

Model	**# Params**	**Scaffold**		
		RMSE	Pearson	Spearman
Sequence-based Methods				
Öztürk et al. [2018]	1.93 M	1.908 ± 0.145	0.384 ± 0.014	0.387 ± 0.016
Bepler and Berger [2019]	48.8 M	1.864 ± 0.009	0.269 ± 0.002	0.285 ± 0.019
Rao et al. [2019]	93.0 M	1.680 ± 0.055	0.487 ± 0.029	0.462 ± 0.051
Elnaggar et al. [2020]	2.4M[1]	1.592 ± 0.009	0.398 ± 0.027	0.409 ± 0.029
Surface-based Methods				
Gainza et al. [2020]	0.62 M	1.583 ± 0.132	0.416 ± 0.111	0.412 ± 0.126
Structure-based Methods				
Hermosilla et al. [2021]	5.80 M	1.592 ± 0.012	0.365 ± 0.024	0.373 ± 0.019
HOLOPROT (●)	1.44 M	1.523 ± 0.028	0.489 ± 0.019	0.491 ± 0.020
HOLOPROT (◆)	1.28 M	$\mathbf{1.516 \pm 0.014}$	$\mathbf{0.491 \pm 0.016}$	$\mathbf{0.493 \pm 0.014}$

● full surface ◆ molecular superpixels

Evaluation metrics. For evaluating different methods, we use three metrics – root mean squared error (RMSE), Pearson correlation coefficient, and Spearman correlation coefficient. We also include the mean and standard deviation across 3 experimental runs.

Results. Table 1 displays the results on protein-ligand binding affinity. HOLOPROT (●, ◆) performs consistently well across different tasks and dataset splits, outperforming all methods on the splits *scaffold* and *identity 60%*. On *identity 30%*, our method outperforms most baselines, while having lower variability across the evaluated metrics. HOLOPROT with molecular superpixels (◆) performs similar to HOLOPROT on the entire surface, with no or little performance loss, suggesting that molecular superpixels capture meaningful biological motifs. We include the models from [Townshend et al., 2020] for completeness, but note that these models were trained only using the protein binding pocket. Binding sites on proteins are often structurally highly conserved regions [Panjkovich and Daura, 2010]. Considering only binding pockets, which vary less between the train and test splits, provides an additional simplification making the task less challenging. All other baselines were tested on the full proteins.

5.2 Enzyme-Catalyzed Reaction Classification

Predicting the reaction class of enzymes without the use of sequence similarity allows for efficient screening of *de novo* proteins, i.e., macromolecules without evolutionary homologs, for catalytic properties [des Jardins et al., 1997]. The architecture of the classification module is described in Equation 3).

[1]The embeddings obtained via Elnaggar et al. [2020] were saved to disk, instead of finetuning the entire pretrained model.

[2]Equivariant neural network (ENN) on binding pocket only.

[3]Graph neural network (GNN) on binding pocket only.

Table 2: **Enzyme-Catalyzed Reaction Classification Results** Comparison of classification accuracy of HOLOPROT against other methods.

Model	**Parameters**	**Reaction Class** Accuracy
Sequence-based Methods		
Hou et al. [2018]	41.7 M	70.9 %
Bepler and Berger [2019]	31.7 M	66.7 %
Rao et al. [2019] (Transformer)	38.4 M	69.8 %
Elnaggar et al. [2020]	420.0 M	72.2 %
Structure-based Methods		
Kipf and Welling [2017]	1.0 M	67.3 %
Derevyanko et al. [2018]	6.0 M	78.8 %
Hermosilla et al. [2021]	9.8 M	**87.2** %
HOLOPROT (●)	0.64 M	77.8 %
HOLOPROT (◆)	0.64 M	78.9 %

● full surface ◆ molecular superpixels

Dataset. Enzyme Commission (EC) numbers constitute an ontological system with the purpose of defining and organizing enzyme functions [Webb, 1992]. The four digits of an EC number are related in a functional hierarchy, where the first level annotates the main enzymatic classes, while the next levels constitute subclasses, e.g. the EC number of the HIV-1 protease is 3.4.23.16. This task aims at predicting the enzyme-catalyzed reaction class of a protein based on according to all four levels of the EC number. We use the same dataset and splits as provided by [Hermosilla et al., 2021], comprising $37,428$ proteins from 384 EC numbers, with $29,215$ instances for training, $2,562$ instances for validation, and $5,651$ for testing. For more details on dataset construction, we refer to Hermosilla et al. [2021, Appendix C].

Baselines. For the classification task, we again compare HOLOPROT against several baselines including sequence-based methods [Hou et al., 2018], methods partially pretrained on millions of sequences [Rao et al., 2019, Bepler and Berger, 2019, Elnaggar et al., 2020] as well as methods utilizing principles of geometric deep learning [Kipf and Welling, 2017, Derevyanko et al., 2018, Hermosilla et al., 2021]. The values for different baselines are taken from [Hermosilla et al., 2021].

Evaluation metric. Model performance is measured via the mean accuracy score.

Results. We report the results of enzyme-catalyzed reaction classification in Table 2. While our method (●, ◆) is unable to outperform the current state-of-the-art method [Hermosilla et al., 2021], we achieve equivalent, if not better results to other methods at a fraction of the parameters used. Molecular superpixels also capture biologically meaningful protein surface motifs, as evidenced by a small increase in the overall classification performance.

5.3 Ablation Studies

To further evaluate the contribution of HOLOPROT to learning multi-scale protein representations, we conduct several ablation studies. First, we analyze if the performance of the multi-scale model outperforms its isolated components, i.e. when using only structure or surface representation for subsequent downstream tasks. The second ablation axis analyzes the construction of molecular superpixel representations. Besides computing summary features for each molecular superpixel as described in Section 4, we learn patch representations via a MPN on the superpixel graph. The ablation study were conducted on both tasks, ligand binding affinity (Section 5.1) and enzyme catalytic function classification (Section 5.2).

As displayed in Table 3, HOLOPROT with (◆) and without molecular superpixels (●) improve over the performance of structure and surface representations. Further, the results of the ablation study clearly show that different protein scales are more relevant for particular downstream tasks, e.g., predicting the enzyme-catalyzed reaction class from surface only results in poor performance. We further see no

Table 3: **Ablation Studies Results** Evaluation of architectural design choices of HOLOPROT by analyzing the performance of its individual components as well as feature summarization of molecular superpixels.

Model	**Ligand Binding Affinity** Sequence Identity (30 %)			**Enzyme Class**
	RMSE	Pearson	Spearman	Accuracy
Structure	1.476 ± 0.027	0.51 ± 0.029	0.503 ± 0.027	74.2 %
Surface	1.482 ± 0.015	$\mathbf{0.512} \pm \mathbf{0.022}$	$\mathbf{0.505} \pm \mathbf{0.017}$	28.6 %
HOLOPROT (●)	$\mathbf{1.464} \pm \mathbf{0.006}$	0.509 ± 0.002	0.500 ± 0.005	77.8 %
HOLOPROT (◆)	1.491 ± 0.004	0.491 ± 0.014	0.482 ± 0.017	**78.9** %
HOLOPROT (■)	1.491 ± 0.027	0.503 ± 0.005	0.492 ± 0.004	75.7 %

● full surface ◆ molecular superpixels ■ molecular superpixel with MPN

improvement in applying a MPN within a molecular superpixel (■) over using summary features (◆). Further ablation studies are presented in Appendix B.1.

5.4 Limitations

Despite the reported success of HOLOPROT, our method faces some limitations. First, HOLOPROT relies on existing protein structures and the corresponding generated surface manifolds. However, protein sequence data still remains the most abundant data source, and in protein design, conformations of mutated macromolecules are unknown. This limitation could however be partly remedied, (i.) by the recent advancements in protein structure prediction [Senior et al., 2020, Jumper et al., 2021, AlphaFold] [Baek et al., 2021, RoseTTAFold] and protein structure determination methods such as cryo-electron microscopy [Callaway, 2020], and (ii.) by utilizing homology modeling algorithms on available wild type structures for mutant analysis [Schymkowitz et al., 2005]. Second, our method requires precomputed surface meshes, resulting in an additional preprocessing step before deploying HOLOPROT to the desired application. This bottleneck can be bypassed by utilizing techniques developed in the concurrent work by Sverrisson et al. [2020], which allow computation and sampling of the molecular surface on-the-fly.

6 Conclusion

In this work, we present a novel multi-scale protein graph construction, HOLOPROT, which integrates finer and coarser representation details of a protein by connecting sequence and structure with surface. We further establish *molecular superpixels*, which capture higher-level fingerprint motifs on the protein surface, improving the memory efficiency of our construction without reducing the overall performance. We validate HOLOPROT's effectiveness and versatility through representative tasks on protein-ligand binding affinity and enzyme-catalyzed reaction class prediction. While being significantly more parameter-efficient, HOLOPROT performs consistently well across different tasks and dataset splits, partly outperforming current state-of-the-art methods. This will potentially be of great benefit and advantage when working with datasets of reduced size, e.g., comprising experiments on mutational fitness of proteins, thus opening up new possibilities within protein engineering and design, which we leave for future work.

Acknowledgments

This project received funding from the Swiss National Science Foundation under the National Center of Competence in Research (NCCR) Catalysis under grant agreement 51NF40 180544. Moreover, we thank Mojmír Mutný and Clemens Isert for their valuable feedback.

References

M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al. Accurate prediction of protein structures and interactions using

a three-track neural network. *Science*, 373(6557), 2021.

N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon. Electrostatics of nanosystems: application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences*, 98(18):10037–10041, 2001.

F. Baldassarre, D. Menéndez Hurtado, A. Elofsson, and H. Azizpour. GraphQA: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 37(3), 2021.

G. W. Bemis and M. A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

T. Bepler and B. Berger. Learning Protein Sequence Embeddings using Information From Structure. In *International Conference on Learning Representations (ICLR)*, 2019.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1), 2000.

L. Biewald. Experiment Tracking with Weights and Biases, 2020. URL `https://www.wandb.com/`.

Blender Online Community. Blender - a 3d modelling and rendering package, 2018. URL `http://www.blender.org`.

A. Brik and C.-H. Wong. HIV-1 protease: mechanism and drug discovery. *Organic & Biomolecular Chemistry*, 1(1), 2003.

E. Callaway. 'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures. *Nature*, 2020.

V. Cantoni, R. Gatti, and L. Lombardi. Segmentation of SES for Protein Structure Analysis. In *Bioinformatics*, 2010.

V. Cantoni, R. Gatti, and L. Lombardi. 3D Protein Surface Segmentation through Mathematical Morphology. In *International Joint Conference on Biomedical Engineering Systems and Technologies*. Springer, 2011.

D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 2002.

M. L. Connolly. Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 221(4612), 1983.

A. Dalkiran, A. S. Rifaioglu, M. J. Martin, R. Cetin-Atalay, V. Atalay, and T. Doğan. ECPred: a tool for the prediction of the enzymatic functions of protein sequences based on the EC nomenclature. *BMC Bioinformatics*, 19(1), 2018.

G. Derevyanko, S. Grudinin, Y. Bengio, and G. Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23), 2018.

M. des Jardins, P. D. Karp, M. Krummenacker, T. J. Lee, and C. A. Ouzounis. Prediction of enzyme classification from protein sequence without the use of sequence similarity. In *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, volume 5, 1997.

P. D. Dobson and A. J. Doig. Predicting Enzyme Class From Protein Structure Without Alignments. *Journal of Molecular Biology*, 345(1), 2005.

T. J. Dolinsky, P. Czodrowski, H. Li, J. E. Nielsen, J. H. Jensen, G. Klebe, and N. A. Baker. Pdb2pqr: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic acids research*, 35(suppl_2):W522–W525, 2007.

A. Elnaggar, M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, D. Bhowmik, et al. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv Preprint*, 2020.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2), 2004.

M. Fessenden. Protein maps chart the causes of disease. *Nature*, 549(7671), 2017.

M. Fey and J. E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric. *International Conference on Learning Representations (ICLR), Workshop Track*, 2019.

A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur. Protein Interface Prediction using Graph Convolutional Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

P. Gainza, F. Sverrisson, F. Monti, E. Rodola, D. Boscaini, M. Bronstein, and B. Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2), 2020.

J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning (ICML)*, 2017.

V. Gligorijevic, P. D. Renfrew, T. Kosciolek, J. K. Leman, K. Cho, T. Vatanen, D. Berenberg, B. Taylor, I. M. Fisk, R. J. Xavier, R. Knight, and R. Bonneau. Structure-Based Function Prediction using Graph Convolutional Networks. *Nature Communications*, 12(1), 2021.

A. Golovin and K. Henrick. MSDmotif: exploring protein sites and motifs. *BMC Bioinformatics*, 9 (1), 2008.

J. Greer and B. L. Bush. Macromolecular shape and surface maps by solvent exclusion. *Proceedings of the National academy of Sciences*, 75(1), 1978.

R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. MeshCNN: A Network with an Edge. In *ACM Transactions on Graphics (TOG)*, volume 38, 2019.

P. Hermosilla, M. Schäfer, M. Lang, G. Fackelmann, P.-P. Vázquez, B. Kozlikova, M. Krone, T. Ritschel, and T. Ropinski. Intrinsic-Extrinsic Convolution and Pooling for Learning on 3D Protein Structures. In *International Conference on Learning Representations (ICLR)*, 2021.

J. Hou, B. Adhikari, and J. Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8), 2018.

J. Jiménez, M. Skalic, G. Martinez-Rosell, and G. De Fabritiis. K_{DEEP}: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, 58(2), 2018.

W. Jin, R. Barzilay, and T. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning*, pages 4839–4848. PMLR, 2020.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), 2021.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12): 2577–2637, 1983.

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

P. Kohli, P. H. Torr, et al. Robust Higher Order Potentials for Enforcing Label Consistency. *International Conference on Computer Vision (ICCV)*, 82(3), 2009.

G. Landrum. RDKit: Open-Source Cheminformatics Software. 2016. URL `https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4`.

B. Lee and F. M. Richards. The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology*, 55(3), 1971.

T. Lei, W. Jin, R. Barzilay, and T. Jaakkola. Deriving Neural Architectures from Sequence and Graph Kernels. In *International Conference on Machine Learning (ICML)*, 2017.

M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy Rate Superpixel Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Z. Liu, M. Su, L. Han, J. Liu, Q. Yang, Y. Li, and R. Wang. Forging the Basis for Developing Protein–Ligand Interaction Scoring Functions. *Accounts of Chemical Research*, 50(2), 2017.

G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering Human Body Configurations: Combining Segmentation and Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2004.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical programming*, 14(1), 1978.

H. Öztürk, A. Özgür, and E. Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17), 2018.

A. Panjkovich and X. Daura. Assessing the structural conservation of protein pockets to study functional and allosteric sites: implications for drug discovery. *BMC structural biology*, 10(1): 1–14, 2010.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, and D. R. Koes. Protein–ligand scoring with convolutional neural networks. *Journal of chemical information and modeling*, 57(4):942–957, 2017.

R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, and Y. S. Song. Evaluating Protein Transfer Learning with TAPE. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

X. Ren and J. Malik. Learning a Classification Model for Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2003.

M. F. Sanner, A. J. Olson, and J.-C. Spehner. Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers*, 38(3), 1996.

J. Schymkowitz, J. Borg, F. Stricher, R. Nys, F. Rousseau, and L. Serrano. The FoldX web server: an online force field. *Nucleic Acids Research*, 33, 2005.

A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792), 2020.

M. M. Stepniewska-Dziubinska, P. Zielenkiewicz, and P. Siedlecki. Improving detection of protein-ligand binding sites with 3d segmentation. *Scientific Reports*, 10(1), 2020.

D. Stutz, A. Hermans, and B. Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166, 2018.

F. Sverrisson, J. Feydy, B. Correia, and M. Bronstein. Fast end-to-end learning on protein surfaces. *bioRxiv*, 2020.

R. Townshend, R. Bedi, P. Suriana, and R. Dror. End-to-End Learning on 3D Protein Structure for Interface Prediction. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

R. J. Townshend, M. Vögele, P. Suriana, A. Derry, A. Powers, Y. Laloudakis, S. Balachandar, B. Anderson, S. Eismann, R. Kondor, et al. ATOM3D: Tasks On Molecules in Three Dimensions. *NeurIPS Workshop of Learning Meaningful Representations of Life (LMRL)*, 2020.

L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Computer Architecture Letters*, 13(06), 1991.

E. C. Webb. *Enzyme Nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.* Academic Press, 1992.

K. K. Yang, Z. Wu, and F. H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8), 2019.

Appendix

A Message Passing Network

We utilize the Weisfeiler-Lehmann network (WLN) proposed in [Lei et al., 2017] as our base message passing network. This network builds a neural equivalent of the Weisfeiler-Lehmann test for comparing graphs. For clarity, we describe the network here. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Given a node $v \in \mathcal{G}$ with neighbors $N(v)$, node features $\mathbf{f}_v$ and edge features $\mathbf{f}_{uv}$ for edge $(v, u) \in \mathcal{E}$, the WLN message passing step follows as,

$$\mathbf{m}_v^{(l)} = \tau(\mathbf{U_1}\mathbf{m}_v^{(l-1)} + \mathbf{U_2} \sum_{u \in N(v)} \tau(\mathbf{V}[\mathbf{f}_u, \mathbf{f}_v])) \quad (1 \leq l \leq L) \tag{4}$$

where $\tau(\cdot)$ could be any non-linear function, and L is the total number of message passing steps, $\mathbf{h}_v^{(0)} = \mathbf{f}_v$. The final representations for each node arise from mimicking the set comparison function in the WL isomorphism test, yielding

$$\mathbf{h}_v = \sum_{u \in N(v)} \mathbf{W}^{(0)}\mathbf{m}_u^{(L)} \odot \mathbf{W}^{(1)}\mathbf{f}_{uv} \odot \mathbf{W}^{(2)}\mathbf{m}_v^{(L)} \tag{5}$$

B Multi-Scale Protein Representations

B.1 Ablation Studies

Table 4 shows ablation study results for the *identity 60%* and *scaffold* splits for PDBBIND dataset. The table will be updated with values of HOLOPROT (molecular superpixels with MPN) setting once the results for the same are available.

Table 4: **Results of the Ablation Studies** Evaluation of architectural design choices of HOLOPROT by analyzing the performance of its individual components as well as feature summarization of molecular superpixels.

Model	**Ligand Binding Affinity**					
	Sequence Identity (60 %)			Scaffold		
	RMSE	Pearson	Spearman	RMSE	Pearson	Spearman
Structure	1.378 ± 0.027	0.738 ± 0.014	0.730 ± 0.009	1.521 ± 0.023	0.485 ± 0.015	0.492 ± 0.013
Surface	1.418 ± 0.014	0.719 ± 0.005	0.714 ± 0.004	1.558 ± 0.125	0.428 ± 0.159	0.429 ± 0.181
HOLOPROT (●)	**1.365 ± 0.038**	**0.749 ± 0.014**	**0.742 ± 0.011**	1.523 ± 0.028	0.489 ± 0.019	0.491 ± 0.021
HOLOPROT (◆)	1.473 ± 0.024	0.667 ± 0.011	0.675 ± 0.019	**1.517 ± 0.014**	**0.491 ± 0.016**	**0.493 ± 0.014**

● full surface ◆ molecular superpixels

C Superpixels on Molecular Surfaces

C.1 Computing Molecular Surfaces

Shape and surface of proteins determines their molecular interactions and thus, accurate computation of macromolecular surfaces from the provided atom point clouds is essential for elucidating their biological roles in physiological processes. A variety of methods have been proposed to compute macromolecular surfaces. *Van der Waals surfaces* is the simplest surface constructed via the topological boundary of the set of atom spheres, each of van der Waals radius of the constituent atom. However, as most of the van der Waals surface is buried in the interior of large molecules, Lee and Richards [1971] defined the solvent-accessible surface (SAS), determined by the area traced out by the center of a probe sphere as it is rolled over the van der Waals surface. Greer and Bush [1978] proposed smooth solvent-excluded surfaces (SES, or *molecular surface*) of a molecule [Connolly, 1983] defined as the boundary of the union of all possible probes having no intersection with the molecule. In this work, we utilize existing algorithm `MSMS` (Michel Sanner's Molecular Surface)

computing triangulated representations of the *molecular surface* relying on a reduced surface [Sanner et al., 1996].

Details on Surface Preparation All proteins were triangulated using the MSMS with a hydrogen density of 3.0 and a water probe radius of 1.5. The meshes are downsampled using BLENDER [Blender Online Community, 2018] to a uniform size of roughly 2600 faces. In practice, we found that this size provided an appropriate balance between maintaining detail and memory consumption during preprocessing. Geometric and chemical features were computed directly on the protein mesh.

C.2 Examples of Molecular Superpixels

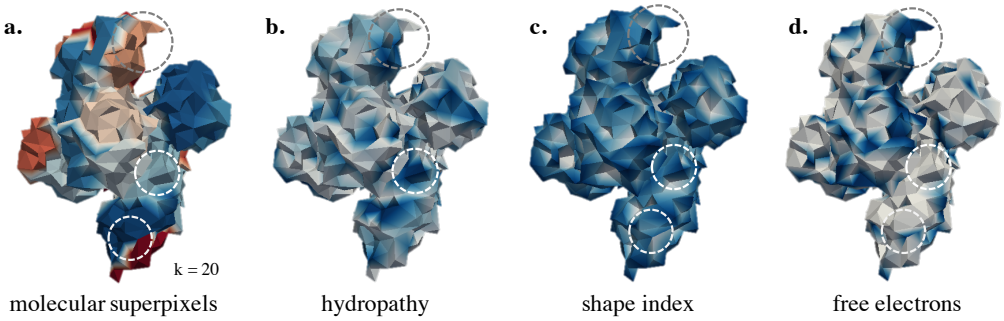

Figure 3: **Molecular Superpixels and Surface Features of the Hepatitis C Virus Helicase Inhibitor** (`PDB ID: 4OKS`)**.** Molecular superpixels, indicated by different colors ($k = 20$), and the corresponding surface features, i.e., **b.** hydropathy, **c.** shape index, and **d.** free electrons. As highlighted, molecular superpixels are spatially compact and overlap with surface regions dominated by single features such as hydrophobic patches while capturing coherent areas across all surface features. Protein complex contains 867 residues.

Additional examples of molecular superpixels and their overlap with different surface features are shown in Figure 3 and Figure 4.

D Experimental Details

Our model is implemented in `PyTorch` [Paszke et al., 2019] using the `PyTorch Geometric` library [Fey and Lenssen, 2019]. We use the open-source software `RDKit` [Landrum, 2016]. We used `W&B` [Biewald, 2020] for experiment tracking.

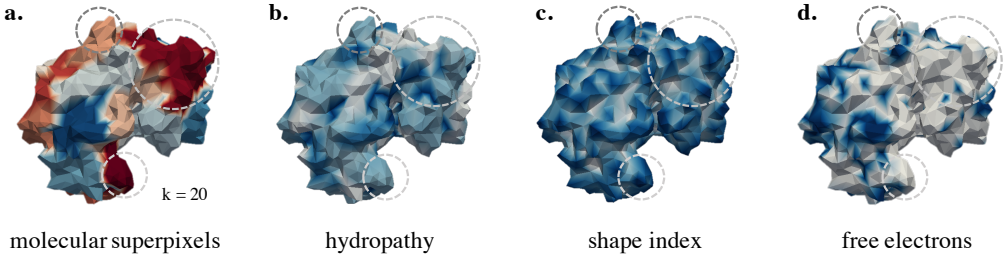

Figure 4: **Molecular Superpixels and Surface Features of Endothia Aspartic Proteinase** (`PDB ID: 1EPO`)**.** Molecular superpixels, indicated by different colors ($k = 20$), and the corresponding surface features, i.e., **b.** hydropathy, **c.** shape index, and **d.** free electrons. As highlighted, molecular superpixels are spatially compact and overlap with surface regions dominated by single features such as hydrophobic patches while capturing coherent areas across all surface features. Protein complex contains 330 residues.

D.1 Features

D.1.1 Surface Layer

We represent the surface layer as a graph $\mathcal{G}_\mathcal{S}$ where, for each node $u_\mathcal{S}$, we compute 4 geometric and chemical features – shape index, free electrons and proton donors, hydropathy and poisson-boltzmann electrostatics. These features are computed using code from [Gainza et al., 2020], and the binaries APBS [Baker et al., 2001], PDB2PQR [Dolinsky et al., 2007] and multivalue (provided within the APBS suite). We refer to [Gainza et al., 2020] for more details.

Two nodes share an edge if they are part of the same triangulation, and an edge is a part of two triangular faces. We compute 7 edge features – the dihedral angle between the two faces, the inner angles (one for each face) opposite to the edge, two edge-length ratios, where the edge ratio is between the length of the edge and the perpendicular (dotted) line for each adjacent face. These features were taken from [Hanocka et al., 2019]. We also include the distance between the surface nodes comprising the edge and the angle between the normals at those nodes.

D.1.2 Structure Layer

We represent the structure layer as a graph $\mathcal{G}_\mathcal{B}$ where the nodes $u_\mathcal{B}$ are the amino acid residues, and the edges occur between two amino acids within a certain distance threshold. We use the following node and edge features,

Node Feature	Count	One-Hot	Possible Values
Residue Name	23	Yes	ALA, GLY etc.
Secondary structure the residue is part of	8	Yes	H, G, I, E, B, T, C, unk
Solvent Accessible Surface of the residue	1	No	-
Residue hydrophobicity	1	No	-

As edge features, we use the angle between two residues and the distance between their C_α atoms. To compute the secondary structure, we use the DSSP binary [Kabsch and Sander, 1983].

D.1.3 Ligand Molecules

We represent the ligand molecule as a graph $\mathcal{G}_\mathcal{L}$ with the following node and edge features,

Node Feature	Count	One-Hot	Possible Values
Atom symbol	65	Yes	C, N, O etc.
Atom degree	10	Yes	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Implicit valence of the atom	6	Yes	0, 1, 2, 3, 4, 5
Explicit valence of the atom	6	Yes	1, 2, 3, 4, 5, 6
Part of an aromatic ring	1	No	0, 1

Edge Feature	Count	One-Hot	Possible Values
Bond type	4	Yes	Single, Double, Triple, Aromatic
Whether bond is conjugated	1	No	0, 1
Whether bond is part of ring	1	No	0, 1

D.2 Datasets

Protein-Ligand Binding Affinity We use the refined subset of the 2019 version of PDBBind [Liu et al., 2017] and evaluate our model on 3 splits – *scaffold*, *identity 60%* and *identity 30%*. Two of these splits (*identity 60%* and *identity 30%*) are based on sequence identity, with sequences in the test set not having more than a 30% or 60% to sequences in the training set. We use the same splits as provided by [Townshend et al., 2020], and refer the reader to the same for more details on their construction. The scaffold split is prepared by computing the Bemis-Murcko scaffold [Bemis and Murcko, 1996] using `RDKit` for each molecule, and splitting the molecules such that molecules with rare or unseen scaffolds are part of the test set.

Enzyme-Catalyzed Reaction Classification We use the PDB files and splits provided by Hermosilla et al. [2021] for this task. For more details on the dataset construction, we refer the reader to Hermosilla et al. [2021, § C].

D.3 Network Architectures

Hyperparameter Tuning For protein-ligand binding affinity prediction, we performed a hyperparameter sweep over the hidden dimensions of the surface $(150, 200, 300)$ and structure layers $(150, 200, 300)$, and the hidden dimensions of the MLP $(512, 256, [512, 256])$. For the enzyme-catalyzed reaction classification, given time constraints, our hyperparameter tuning was restricted to the learning rates $0.001, 0.0005, 0.0001$ and hidden layer activations $(\mathrm{ReLU}, \mathrm{LeakyReLU})$.

For all HOLOPROT models, we use the Adam optimizer for training and clip gradients to a maximum norm of 10.0.

D.3.1 Protein-Ligand Binding Affinity

For HOLOPROT with the full surface, the surface and structure layer MPNs have hidden dimensions of 150 and 200 with message passing steps of 6 and 5 respectively. The affinity prediction MLP has a single hidden layer of dimension 512. For HOLOPROT with molecular superpixels, the surface and structure layer MPNs have hidden dimensions of 150 and 300, with 5 message passing steps. The affinity prediction MLP has a single hidden layer of dimension 256. For both models, the ligand MPN has a hidden layer dimension of 300, with 4 message passing steps. We use the ReLU activation function. Starting with an initial learning rate of 0.001, we apply a learning rate decay of 0.9 based on a validation RMSE plateau, with an improvement threshold of 0.01 and a patience of 5. The HOLOPROT model for full surface has 1.44M parameters, while the HOLOPROT model with molecular superpixels has 1.76M parameter.

D.3.2 Enzyme-Catalyzed Reaction Classification

Both HOLOPROT models have a hidden dimension of 150 for both the surface and structure layers, with 4 and 5 message passing steps. The classification MLP has a single hidden layer of dimension 512. We use the $\mathrm{LeakyReLU}$ activation function, and apply dropout with probability 0.15 for each message passing step, and 0.3 for the classification MLP. Starting with an initial learning rate of 0.0005, we apply a learning rate decay of 0.6 based on a validation accuracy plateau, with an improvement threshold of 0.01 and a patience of 10. Both models have roughly 0.64M parameters. We ran HOLOPROT with more parameters (5.2 M) *without* hyperparameter search due to computational restrictions, resulting in small improvement of the overall performance (i.e., an accuracy of 79.2% on the enzyme-catalyzed reaction classification task).

D.3.3 Baselines

For protein-ligand binding affinity prediction, we use the provided code for different baselines and extend them as necessary for the task. For enzyme-catalyzed reaction classification, we use the baseline values from [Hermosilla et al., 2021].

Across all models for protein-ligand binding affinity prediction, we compute representations for proteins and ligands and concatenate them and use that as input for the MLP. For the models described in [Townshend et al., 2020], we use the reported values. Wherever possible, we restrict the ligand embedding dimension to be 300 consistent with our experiments. The details for the remaining baselines are as follows,

Öztürk et al. [2018] We downloaded the code from the official repository. The authors do not use a separate validation set, but instead use a cross-validation strategy. We combine the training and validation sets and then perform 5-fold cross-validation. The authors allow specification of hyperparameters for the number of filters (32, 64), the size of ligand sequence filters (4, 8, 12, 36), and size of protein sequence filters (4, 8, 12, 36). We use a default batch size of 64, and the default learning rate, and train the model for 100 epochs. We also note that the model typically undergoes early stopping around epoch 80. The best performing model occurred with 32 filters, with a filter size of 4 for ligand sequences, and 8 for protein sequences.

Bepler and Berger [2019] We downloaded the code and pretrained models from the official repository and embed the ligand sequence with a bidirectional LSTM. For embedding the protein, we use the default values as from the pretrained model. For the ligand, we use an input dimension of 512, and a LSTM hidden dimension of 512, with a final embedding dimension of 100, similar to the protein. Our hyperparameter tuning is restricted to ligand input dimensions $(512, 256)$, LSTM hidden dimensions $(512, 256)$ and the MLP hidden dimensions $(512, 256, [512, 256])$. We use a batch size of 32 and learning rate of 0.00001 for the pretrained model, and 0.001 for the remaining parameters.

Rao et al. [2019] We downloaded the code and pretrained models from the official repository. We represent the ligand as a sequence and embed it using a Transformer, with an embedding dimension of 300 and an intermediate size of 512. We use a 2-layer MLP with hidden dimensions of 512 and 256 and dropout probabilities of 0.2 to predict the binding affinity after concatenating the protein and ligand embeddings. For training, we use a learning rate of 0.0001 for the pretrained model parameters and 0.001 for the remaining parameters, and trained the model for 300 epochs. Our hyperparameter sweep was restricted to batch size (default value 32, multiplied and divided by 2 until no improvement), and hidden layer dimensions for the MLP $(512, 256, [512, 256])$.

Gainza et al. [2020] We downloaded the code from the official repository and extended the model for the ligand binding affinity task. For the protein, we use the default values provided. We represent the ligand as a graph and use the same architecture and parameters as our message passing network, with a hidden dimension of 300. Given memory and time constraints, we were unable to perform a hyperparameter sweep.

Hermosilla et al. [2021] We downloaded the code from their official repository. The proteins are embedded with an embedding dimension of 1024. We represent the ligand as a graph and use the same architecture and parameters as our message passing network, with a hidden dimension of 300. We concatenate the protein and ligand embeddings before using it as input for a single-layer MLP with hidden size 512. The model is trained with the default learning rate 0.001 and learning rate decay for 300 epochs. Due to memory constraints, we trained with the default batch size of 8, and performed a hyperparameter sweep for MLP hidden sizes 512, 256, $[512, 256]$, and protein embedding dimensions $1024, 512$.

D.4 Computing Infrastructure

All models were trained on a single NVIDIA 1080Ti GPU. For the PDBBind dataset, all the HOLOPROT models run within 16 hours when trained for 200 epochs, and within 8 hours when trained for 100 epochs. The baseline models [Rao et al., 2019, Hermosilla et al., 2021] take about 40 hours for running 300 epochs, while the remaining baseline models train in under 24 hours. For the enzyme-catalyzed reaction classification dataset, the HOLOPROT models are trained for 24 hours on the NVIDIA 1080Ti GPU after which it is stopped. Typically, the model is trained for 100 to 110 epochs by then, when using a batch size of 10.