PROTSURF: LEARNING TRANSFERABLE RESIDUE-ALIGNED PROTEIN SURFACE REPRESENTATIONS FROM LARGE-SCALE PROTEIN INTERFACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein sequence and structure have emerged as powerful reusable modalities for protein representation learning. In contrast, protein surfaces—which mediate protein-protein interactions—remain underexplored as transferable representations. We introduce ProtSurf, a hierarchical self-supervised framework that formulates protein surface learning as a scalable representation learning problem and learns residue-aligned surface representations from large-scale protein interfaces. ProtSurf models protein surfaces at local and global scales and investigates how protein surface representations benefit from increasing model capacity and pretraining data diversity. We pretrain ProtSurf on approximately 200,000 experimentally determined RCSB protein structures and 1.6 million domain–domain interfaces derived from AlphaFold-predicted structures. We evaluate ProtSurf residue embeddings for binding affinity prediction and achieve strong performance across multiple benchmarks. Beyond affinity prediction, ProtSurf provides, to our knowledge, the first transferable residue-level surface representation learned at scale, establishing protein surfaces as a representation modality alongside sequence and structure.

1 INTRODUCTION

Protein sequence and structure have emerged as powerful modalities for protein representation learning. Large-scale self-supervised models demonstrate that representations learned from diverse protein corpora can transfer across tasks including structure prediction, protein design and functional annotation. In contrast, protein surfaces—which directly mediate protein-protein interaction (PPI) and binding—remain underexplored as a transferable representation modality. Although surface descriptors have been widely used for binding-site prediction, protein–ligand interaction modeling, and protein–protein interface analysis (Gainza et al., 2020), surface information has largely remained a task-specific feature or architectural input rather than a reusable representation in its own right.

Protein surface fingerprint is local and geometric, whereas modern protein models typically operate on residue-indexed representations. Past approaches that have used surface modality produce representations at surface mesh points, or combine surface information with separate atomic or structural representations (Mallet et al., 2023; Wu & Li, 2024; Zhang et al., 2024; Somnath et al., 2021). A reusable surface representation should instead be learned directly from large collections of proteins with diverse interfaces. Residue alignment is particularly beneficial because it places surface information on the same index used by sequence- and structure-based protein models, enabling surface-derived representations to be integrated directly with other residue-indexed modalities (Hayes et al., 2025). Finally, the growing availability of experimentally resolved complexes and AlphaFold-predicted structures (Varadi et al., 2024; Burley et al., 2023) now makes large-scale learning of such representations feasible.

In this work, we introduce **ProtSurf**, a self-supervised framework whose primary contribution is a *residue-aligned protein surface representation* that can be learned at scale and reused by downstream protein learning models. ProtSurf's primary contribution is to formulate protein surface information as a reusable representation modality instead of introducing a new surface descriptor. It first maps local chemical and geometric surface features to each surface-exposed residue and then contextualizes these residue-level representations through a protein-level graph using graph topology-aware attention.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We pre-train ProtSurf on experimentally determined proteins from the RCSB Burley et al. (2023) database and **1.6 million** interacting protein-domain interfaces derived from AlphaFold-predicted structures. We scale the model from 10M to 650M parameters and systematically study the effects of model capacity and pretraining data diversity on downstream affinity prediction, a natural setting for evaluating surface representations. Specifically, we extract residue embeddings from pretrained ProtSurf and evaluate them on SKEMPI Jankauskaitė et al. (2019), HER2 Shanehsazzadeh et al. (2023), and retrospective ranking of experimentally validated mutations against SARS-CoV-2 Yuan et al. (2020). ProtSurf achieves strong performance among the evaluated methods and improves transfer across these diverse settings, providing evidence that protein surface representations can benefit from the same large-scale representation-learning paradigm that has proven effective for sequence and structure.

Our contributions are therefore threefold: (i) a hierarchical framework for learning residue-aligned protein surface representations at scale; (ii) large-scale pretraining on experimentally determined proteins from the RCSB Burley et al. (2023) database and 1.6 million protein interfaces, together with systematic studies of model capacity, pretraining data diversity, and representation dimensionality; and (iii) evidence across multiple protein-interaction benchmarks that the resulting surface representations transfer beyond their self-supervised pretraining objective.

2 Related Work

Protein surface representation learning. MaSIF (Gainza et al., 2020) and later dMaSIF (Sverrisson et al., 2021) established molecular surfaces as useful representations for protein binding and interaction. Subsequent approaches have either incorporated surface descriptors into structure-based architectures (Somnath et al., 2021; Zhang et al., 2024; Mallet et al., 2023) or learned surface representations through reconstruction and pretraining objectives (Wu & Li, 2024; Song et al., 2024; Lee & Kim, 2023). ProtSurf follows the latter direction but targets a different representation: a reusable surface representation aligned with individual protein residues. Table 1 summarizes the key differences.

Relation to AtomSurf and Surface-VQMAE. The closest comparisons are AtomSurf (Mallet et al., 2023) and Surface-VQMAE (Wu & Li, 2024). AtomSurf combines a surface graph with an all-atom structure graph and can generate residue-level embeddings. Surface-VQMAE learns representations of surface patches through masked reconstruction. ProtSurf instead reconstructs both node features and graph connectivity, maps local surface information to residue-indexed representations, and contextualizes these representations through a global protein-level graph. ProtSurf also pretrains on experimentally determined proteins from the RCSB Burley et al. (2023) database and 1.6 million protein interfaces, substantially increasing the scale of surface representation learning.

Residue-aligned surface representations advantage. We view *surface modality and residue alignment* as the key distinctions between ProtSurf and prior surface-based approaches. Separating the surface representation from a parallel backbone or all-atom feature representation allows us to test whether surface information provides complementary information beyond the structural features used by the downstream predictor. This opens the possibility of using these representations in modern multi-modal protein models such as ESM3 (Hayes et al., 2025), SaProt (Su et al., 2023) and ProSST (Li et al., 2024) which organize multiple modalities along a shared residue index.

Binding affinity prediction. Computational approaches to protein–protein binding affinity range from physics-based energy functions such as FoldX Schymkowitz et al. (2005) and Rosetta Alford et al. (2017) to learned predictors based on inverse folding, all-atom graphs, and sequence–structure representations Dauparas et al. (2022); Luo et al. (2023a); Cai et al. (2024). Among recent methods, BA-DDG Jiao et al. (2025) and EBM-DDG Soga et al. (2025) explicitly formulate $\Delta\Delta G$ through the thermodynamic cycle between bound and unbound states, approximating the unbound state using the separated component chains; EBM-DDG additionally introduces a learned structural-energy term to account for mutation-induced conformational changes. Pythia-PPI Tao et al. (2025) and DSSA-PPI Wu et al. (2026) instead represent affinity-specific predictors built on pretrained structural or joint sequence–structure representations. PreFold-dG Park et al. (2026) predicts absolute ΔG for wild-type and mutant complexes from intermediate Boltz-2 representations and obtains $\Delta\Delta G$

by subtraction, enforcing antisymmetry and transitivity. ProtSurf differs from these task-specific formulations by learning residue-aligned molecular-surface representations without affinity labels and keeping the encoder frozen during downstream evaluation, thereby isolating the transferability of the pretrained surface representation.

Table 1: Comparison of representative surface-based protein representation methods. "Output granularity" denotes the index of the learned representation; residue-aligned outputs correspond directly to protein residues.

Method	**Input**	**Pretraining objective**	**Output granularity**
MaSIF	Surface	None (supervised)	Surface
dMaSIF	Surface	None (supervised)	Surface
AtomSurf	Surface + Structure	None (supervised)	Surface + residue-aligned
Surface-VQMAE	Surface	Masked node features	Surface
ProtSurf	Surface	Node features + adjacency	Residue-aligned

3 PROTSURF: LEARNING RESIDUE-ALIGNED SURFACE REPRESENTATIONS

ProtSurf is a hierarchical self-supervised framework for learning residue-aligned protein surface representations that can be transferred across downstream protein-interaction tasks. We use *transfer* to mean that the surface encoder is pretrained on diverse protein interfaces from full protein complexes and protein domains using a self-supervised objective, after which the frozen encoder's residue embeddings are used in downstream evaluations. ProtSurf represents surface-exposed residues only and does not represent buried residues or the full protein backbone. For surface-associated residues, C_α and C_β coordinates are used to establish residue–surface correspondence and to define spatial relationships in the global graph. They are not used to construct a separate backbone feature representation. Figure 1 provides an overview.

3.1 RESIDUE-ALIGNED LOCAL AND GLOBAL SURFACE GRAPHS

At the local scale, each surface-associated residue is represented by a graph describing its surrounding geometric and physicochemical environment. At the global scale, these residue-level representations are connected through a protein graph to provide broader structural context.

Local residue-centered surface graphs. For each local surface graph, we use the geometric and chemical descriptors introduced by MaSIF (Gainza et al., 2020): electrostatic potential, hydrophobicity, hydrogen-bond interaction, shape index, and distance-dependent curvature. We additionally use two residue-anchored geometric features: (i) the distance between each patch centroid and the residue's C_α atom, and (ii) the angle defined by the patch centroid, C_α, and C_β. These features explicitly associate sampled surface observations with their corresponding residue. The C_α and C_β coordinates establish this correspondence; the learned local representation is derived from the protein surface and its associated descriptors. We show in ablation (Appendix F.3) the usefulness of these features in downstream tasks. For each residue, ProtSurf samples a fixed set of surface patches using category-stratified Farthest Point Sampling, prioritizing patches associated with the residue and incorporating neighboring side-chain information. Details of featurization are provided in Appendix A.1.

Global protein graph. The local encoder produces one latent representation for each surface-exposed residue. These representations are contextualized using a global protein graph whose nodes correspond to surface-exposed residues and whose edges are defined from the corresponding residue C_α coordinates. The global module therefore propagates information among surface-derived residue representations while using residue coordinates only to define graph topology. Unlike AtomSurf (Mallet et al., 2023), ProtSurf does not introduce a separate backbone or all-atom feature

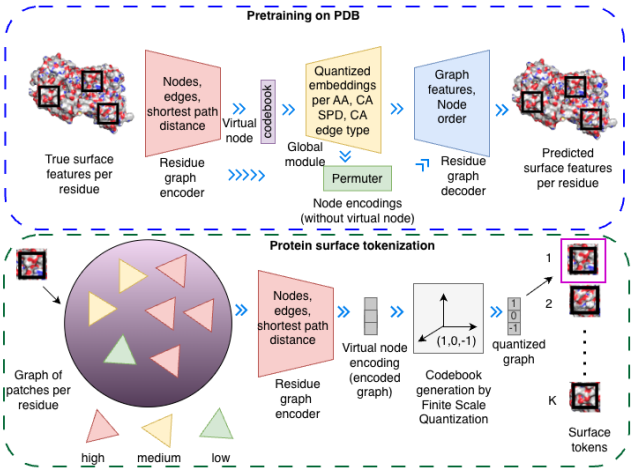

Figure 1: ProtSurf architecture. Local surface residue graphs are encoded and contextualized through a global protein-level module. Residue C_α coordinates define the global graph topology, while learned node representations are derived from molecular surface features.

representation. The C_α coordinates provide the structural scaffold for surface graph construction rather than learned backbone features. Additional details of the global graph construction are provided in Appendix A.2 and A.3.

3.2 Hierarchical surface representation learning

The local and global stages serve complementary roles: the local encoder captures fine-grained geometric and physicochemical relationships within each residue-centered surface graph, while the global module contextualizes these residue-level representations across the protein. Both stages use graph Transformers with learnable relative positional encodings (Park et al., 2022). The relative encodings incorporate graph topology and edge relationships into attention, allowing the same modeling principle to operate at both scales. The full attention formulation is provided in Appendix B.1.

Discrete representation bottleneck. ProtSurf applies finite scalar quantization (FSQ) (Mentzer et al., 2023) to its residue-level representations, producing a discrete code for each surface-associated residue. We retain FSQ as a representational format that may facilitate future token-based multimodal or generative applications; it is not treated as a source of downstream affinity improvements. Details are provided in Appendix B.2.

3.3 Self-supervised reconstruction of surface graphs

ProtSurf is trained to reconstruct local surface graphs from their learned residue-level representations. This provides a self-supervised signal that encourages the representations to preserve both surface properties and their relational structure. Because local surface-graph nodes have no canonical ordering, direct reconstruction requires an alignment between the unordered input graph and the decoder's node ordering. ProtSurf therefore uses a permutation module that learns a differentiable alignment between the two orderings. The decoder receives the globally contextualized residue representation and the learned alignment and reconstructs two aspects of the local graph: node features and adjacency structure. Node reconstruction preserves local geometric and physicochemical information, while adjacency reconstruction preserves relational organization within the surface neighborhood. The full permutation formulation and decoder architecture are provided in Appendix B.3 and Appendix B.4. The node reconstruction weights reflect the association of sampled surface patches with their corresponding residue and are normalized at the protein level. Full definitions of the reconstruction terms, weighting scheme, and permutation regularizer are provided in Appendix B.5.

4 EXPERIMENTS

Pretraining data. Our May 2024 RCSB snapshot contains approximately 200,000 experimentally determined PDB entries, including both single-chain structures and multichain entries. We use each processed entry as a full-structure pretraining example. For multichain entries, we additionally extract each constituent protein chain in its bound conformation, remove the partner chains, and recompute MaSIF surface features. Removing the partners exposes surface regions that are buried in the assembled structure, including interface-forming regions, while retaining the experimentally determined chain coordinates. These chain-derived examples are unlabeled augmented views rather than additional experimentally determined structures. All views derived from the same parent PDB entry are assigned to the same pretraining partition. We also curated 1.6M filtered domain–domain interfaces constructed from the AlphaFold Protein Structure Database (Varadi et al., 2022) using domain annotations from the TED database (Lau et al., 2024). The AFDB predicted structures provide additional diversity in protein interfaces by treating intra-chain domain interactions as training examples, following recent work (Didi et al., 2026). Further details on corpus construction and training are provided in Appendix C and Appendix D.

Benchmark overlap. ProtSurf is pretrained exclusively with a self-supervised surface reconstruction objective. While the May 2024 RCSB source corpus contains all 345 PDB entries represented in the canonical SKEMPI 2.0 release, the pretraining does not use SKEMPI binding-affinity labels; this is benchmark-input overlap and our evaluation measures the utility of large-scale self-supervised surface pretraining rather than strict generalization to structures unseen during pretraining. This evaluation setting is consistent with prior self-supervised protein representation learning, where large structural corpora are used for pretraining without requiring benchmark-specific deduplication Wu et al. (2025); Cai et al. (2024). Following Cai et al. (2024), we evaluate ProtSurf on three binding-affinity prediction benchmarks: SKEMPI v2 (Jankauskaitė et al., 2019), HER2 (Shanehsazzadeh et al., 2023), and SARS-CoV-2 (Yuan et al., 2020). For all datasets, relaxed structures of the mutant sequences were created using the FlexDDG protocol Barlow et al. (2018) and we then ran MaSIF to get the surface features. In all experiments, ProtSurf is used as a frozen encoder to extract residue-level surface representations, which are provided to a downstream GearBind predictor (Cai et al., 2024). GearBind uses all-atom interface graphs and geometric features to predict binding affinity. We choose GearBind to test whether the learned surface representation contains complementary information beyond that captured by full-atom structural features. We intentionally keep the ProtSurf encoder frozen, allowing us to evaluate the utility of the pretrained representation without task-specific adaptation of the surface encoder.

SKEMPI v2 evaluation. For SKEMPI v2, we use the three-fold cross-validation protocol of Luo et al. (2023b), including their random seed to create the folds. Within each training fold, we further split the data into 90% training and 10% validation sets following the evaluation strategy of Soga et al. (2025). This split was also performed at the PDB-complex level, with no complex shared between the training and validation sets. We use the same downstream architecture and projection dimension for every fold and random seed. For each run, the checkpoint with the highest validation Pearson correlation is selected and evaluated once on the corresponding held-out test fold. Test metrics are not used for architecture, checkpoint, or seed selection. We report metrics averaged across the three test folds and five random seeds, and provide the mean and standard deviation across all 15 runs.

Several recent methods report strong results on SKEMPI under evaluation settings that differ materially from ours. Pythia-PPI Tao et al. (2025) uses an auxiliary protein-stability task and self-distillation over approximately 400,000 pseudo-labeled mutations generated across SKEMPI interfaces; it does not report results under our three-fold train/validation/test protocol. PreFold-dG Park et al. (2026) uses the RDE-Network PDB fold assignment, but applies a separate SKEMPI preprocessing and label-calibration pipeline, producing 5,817 examples across 334 complexes, and follows a different training and model-selection procedure. DSSA-PPI Wu et al. (2026) reports its full-SKEMPI result using mutation-level cross-validation, in which mutations from the same complex can occur in both training and test partitions; its complex-level experiments instead use different benchmark subsets and do not report the complete set of metrics used in Table 2. We therefore discuss these methods in Related Work but do not include their reported values in Table 2, since ranking results obtained from different samples, partitions, and supervision settings would not provide a controlled comparison of model performance.

Prediction on HER2 and SARS-COV-2 . Following Cai et al. (2024), we train the downstream predictor on one SKEMPI split using the same architecture and projection dimension as in the SKEMPI evaluation, and evaluate the resulting model on HER2 and SARS-CoV-2. The test datasets are not used for hyperparameter optimization, checkpoint selection, or early stopping.

Table 2: Evaluation of mutant effect prediction ($\Delta\Delta G$) on SKEMPI v2. Pearson and Spearman correlations are higher-is-better; MAE and RMSE are lower-is-better. Best results in **bold**, second best underlined.

Category	Method	Per-Struct.		Overall				
		Pear. ↑	Spear. ↑	Pear. ↑	Spear. ↑	RMSE ↓	MAE ↓	AUROC ↑
Energy	Rosetta	0.3284	0.2988	0.3113	0.3468	1.6173	1.1311	0.6562
	FoldX	0.3789	0.3693	0.3120	0.4071	1.9080	1.3089	0.6582
Sequence	ESM-1v	0.0073	-0.0118	0.1921	0.1572	1.9609	1.3683	0.5414
	PSSM	0.0826	0.0822	0.0159	0.0666	1.9978	1.3895	0.5260
	MSA-T	0.1031	0.0868	0.1173	0.1313	1.9835	1.3816	0.5768
	Tranception	0.1348	0.1236	0.1141	0.1402	2.0382	1.3883	0.5885
Supervised	DDGPred	0.3750	0.3407	0.6580	0.4687	1.4998	1.0821	0.6992
	End2End	0.3873	0.3587	0.6373	0.4882	1.6198	1.1761	0.7172
Pretrain	Surf-VQMAE	0.4694	0.4324	0.6482	0.5611	1.5876	1.1271	0.7469
	Prompt-DDG	0.4369	0.3904	0.6661	0.5626	1.5415	1.0989	0.7369
	AtomSurf	0.2321	0.2192	0.3593	0.3168	1.8210	1.5091	0.6019
	GearBind+PT	0.4895	0.4340	0.6760	0.5250	1.6110	1.1150	0.7510
	BA-DDG	0.5266	0.5040	0.6469	0.6034	1.6650	1.1942	0.7533
	EBM-DDG[†]	0.5472	0.5076	0.7140	**0.6526**	1.4436	**1.0102**	**0.7866**
	ProtSurf (650M)	**0.5987**	**0.5260**	**0.7612**	0.6240	**1.4131**	1.0119	0.7701

[†] We report numbers from the EBM-DDG paper with their dedicated train/validation/test splits when available.

5 RESULTS

We evaluate whether ProtSurf provides a transferable surface representation for downstream protein modeling. Specifically, we test whether a pretrained and frozen ProtSurf encoder can provide useful residue-level representations across multiple downstream protein-interaction settings without adapting the surface encoder to the evaluation task. For previously published methods, we report benchmark results from their peer-reviewed publications for comparison. For SurfVQMAE (Wu & Li, 2024), we use the results reported in the original publication because pretrained model weights were not publicly available and we were unable to reproduce the method. For classical methods whose results are reported in the EBM-DDG paper (Soga et al., 2025), we use the published results under its train/validation/test split whenever available.

5.1 PERFORMANCE ON MUTATION-INDUCED BINDING AFFINITY CHANGE PREDICTION (SKEMPI)

ProtSurf obtains the highest per-structure Pearson and Spearman correlations, the highest overall Pearson correlation, and the lowest RMSE among the compared methods shown in Table 2. Adding frozen ProtSurf embeddings to GearBind increases overall Pearson correlation from 0.6760 to 0.7612 (*GearBind+PT in Table 2*). This corresponds to a 0.0852 absolute improvement in overall Pearson correlation over GearBind+PT, obtained without fine-tuning the ProtSurf encoder and using only a 128-dimensional residue representation. The result supports the hypothesis that surface-derived information provides complementary signal to the all-atom structural features already used by GearBind. Note that the GearBind model parameters are approximately 1.8M parameters and the residue embeddings from ProtSurf (128) add negligible parameters to the model. Full results with all methods are shown in Table 8 (Appendix E.1) and metrics averaged across seeds with standard deviation are provided in Table 9 in Appendix E.2.

5.2 PREDICTION ON HER2

We evaluate the 650M ProtSurf model on 419 substitution-only trastuzumab variants with measured HER2 binding Shanehsazzadeh et al. (2023). ProtSurf achieves the highest Pearson and Spearman correlations among the compared methods (see Table 4), with Pearson **0.534** and Spearman **0.536**. Compared with GearBind+Pretrained, ProtSurf improves both Pearson and Spearman correlation by 0.019, indicating that the benefit of the surface representation extends beyond SKEMPI to an independent antibody-binding benchmark. Evaluation of the smaller ProtSurf models on HER2 are shown in Table 10 in Appendix E.3.

5.3 RETROSPECTIVE MUTATION RANKING FOR SARS-COV-2 ANTIBODY OPTIMIZATION

Correlation against measured affinities is an indirect proxy for how affinity prediction models are used in practice. In an affinity maturation campaign, only a limited number of candidates can typically be synthesized, making the ability to prioritize affinity-enhancing mutations particularly important. We retrospectively rank five previously validated affinity-enhancing CR3022 mutations among 494 heavy-chain CDR single-substitution candidates. We took the same 650M model that was evaluated on HER2 for this dataset as well. ProtSurf places four of the five validated mutations in the top 10% and obtains a mean percentile rank of 8.62% (Table 3). Because only five validated positives are available, we treat this result as a focused ranking case study. ProtSurf also achieves the lowest mean percentile rank among the methods, while placing four of the five validated mutations within the top 10%.

Table 3: Retrospective ranking of five experimentally validated affinity-enhancing mutations (Shan et al., 2022) among 494 heavy-chain CDR single-point variants, following the protocol of Luo et al. (2023b). Values are percentile ranks of each mutation in the predicted $\Delta\Delta G$ ordering (lower is better). No SARS-CoV-2 affinity labels are used during training. Best in **bold**, second best underlined.

Method	TH31W	AH53F	NH57L	RH103M	LH104F	Average ↓
MIF-Network	24.49%	4.05%	6.48%	80.36%	36.23%	30.32%
RDE-Network	**1.62%**	2.02%	20.65%	61.54%	5.47%	18.26%
DiffAffinity	7.28%	3.64%	18.82%	81.78%	10.93%	24.49%
Prompt-DDG	2.02%	6.88%	3.24%	34.81%	6.48%	10.69%
MSM-Mut	6.48%	10.12%	16.19%	19.23%	20.04%	14.41%
BA-DDG	5.26%	15.58%	**2.22%**	40.28%	7.69%	14.21%
EBM-DDG	12.96%	24.70%	10.73%	8.91%	2.23%	11.90%
ProtSurf (650M)	7.69%	**1.82%**	22.06%	**6.88%**	**4.66%**	**8.62%**

Table 4: Correlation on HER2 $\Delta\Delta G$ prediction. Pearson and Spearman correlations are higher-is-better. Best results in **bold**.

Method	Pearson ↑	Spearman ↑
FoldX	0.246	0.329
RDE-Net.	0.465	0.481
Bind-ddG	0.387	0.388
BA-DDG	0.326	0.362
AtomSurf	0.100	0.071
GearBind+Pretrained	0.515	0.517
ProtSurf (650M)	**0.534**	**0.536**

5.4 SCALING SURFACE REPRESENTATION LEARNING

We next examine how ProtSurf performance changes with model capacity and pretraining data. As shown in Table 5 and Figure 6 in Appendix F, downstream Pearson correlation increases monotonically from 0.6856 at 10M parameters to 0.7612 at 650M, while per-structure Pearson correlation

increases from 0.5489 to 0.5987 and pretraining reconstruction loss decreases. The parallel improvement in reconstruction and downstream performance is consistent with increasing model capacity producing progressively more useful surface representations for affinity prediction. In addition, the residue embedding dimension does not increase with increase in model size. This was a design choice based on FoldToken Gao et al. (2025), which uses an embedding dimension of 128, and shows that larger models produce richer representations without increasing the output dimensionality.

Pretraining data scale and composition also affect downstream performance. Models trained on diverse interface datasets achieve lower reconstruction error and improved downstream performance compared with models trained on smaller datasets. Adding 1.6M protein-protein domain interaction data to the RCSB proteins improves Pearson correlation by 0.051, while increasing model capacity from 10M to 650M improves it by 0.075. These results indicate that both model capacity and exposure to more diverse pretraining data contribute to the quality and downstream utility of the learned surface representations (Appendix F.5).

Table 5: Effect of model capacity and pretraining data scale. Pearson and Spearman are higher-is-better; MAE and RMSE are lower-is-better. Best results in **bold**.

Model size	Per-Struct.		Overall				
	Pear. ↑	Spear. ↑	Pear. ↑	Spear. ↑	RMSE ↓	MAE ↓	AUROC ↑
10M	0.5489	0.4625	0.6856	0.5450	1.5525	1.1853	0.7193
60M	0.5751	0.4989	0.7033	0.5686	1.4969	1.1402	0.7320
150M	0.5883	0.5048	0.7344	0.5931	1.4907	1.0910	0.7483
220M	0.5902	0.5120	0.7419	0.6078	1.4727	1.0461	0.7632
650M	**0.5987**	**0.5260**	**0.7612**	**0.6240**	**1.4131**	**1.0119**	**0.7701**

6 Ablation Studies

We conduct ablations to isolate the contributions of ProtSurf's key architectural and representation choices, including hierarchical global modeling, residue-anchored geometric features, surface adjacency reconstruction, and discrete surface tokenization. Unless otherwise specified, we evaluate downstream binding-affinity prediction on SKEMPI v2 using the smaller model variants. Because the scaling trends are consistent through 650M, we perform detailed architectural ablations at smaller model sizes while evaluating the final model at full scale. Further experimental details are provided in Appendix F.

Table 6 summarizes the main ablations and scaling controls. Global contextualization provides the largest improvement among the architectural variants: adding a global module stage to the local model improves Pearson correlation by 0.041. Removing the residue-anchored geometric features produces the largest degradation overall (-0.062), highlighting the importance of explicitly relating local surface observations to their associated residues. Removing permutation-based adjacency reconstruction also reduces performance (-0.022), supporting the role of relational structure in the self-supervised objective. The placement of global contextualization matters as well, with the decoder-stage variant outperforming alternatives that apply global modeling before quantization or use a fully global formulation (details in Appendix F.1).

The remaining ablations help distinguish representation quality from representation format. Changing the residue embedding dimension from 128 to either 64 or 256 decreases performance (-0.015 and -0.030, respectively), indicating that the scaling gains are not simply explained by increasing the dimensionality of the final residue representation. In contrast, removing FSQ slightly improves Pearson correlation by 0.006 ($0.7033 \rightarrow 0.7094$ at 60M). Thus FSQ does not produce the downstream accuracy improvement; rather, it provides discrete surface tokens at little observed cost to transfer performance, supporting future integration with token-based protein language models. Detailed results for the embedding-dimension, geometric-feature, adjacency-reconstruction, and FSQ ablations are provided in Appendix F.6, Appendix F.3, Appendix F.4, and Appendix F.2, respectively.

Overall, the ablations identify two properties as particularly important to ProtSurf's learned representation: residue anchoring, which connects local surface observations to specific surface-associated residues, and local-to-global contextualization, which allows these residue-aligned surface represen-

tations to incorporate protein-level spatial context. The results also separate these representational choices from FSQ: while the architectural and geometric components improve downstream performance, FSQ primarily determines how the learned residue representations are encoded and exposed for downstream or future multimodal use.

Table 6: Ablation and scaling studies on SKEMPI v2. Values report changes in Pearson correlation relative to the corresponding reference configuration; positive values indicate improvement.

(a) Architecture and representation

Ablation	Δ Pearson $\uparrow$
Local $\rightarrow$ Local + Global	$+0.041$
Global before quantization	$+0.016$
Fully global variant	$+0.019$
Remove FSQ	$+0.006$
Remove permutation module	-0.022
Remove geometric anchor features	-0.062

(b) Scaling and controls

Comparison	Δ Pearson $\uparrow$
Model scaling: 10M $\rightarrow$ 650M	$+0.075$
Pretraining data: 200k $\rightarrow$ 1.6M	$+0.051$
Embedding dimension: $128 \rightarrow 64$	-0.015
Embedding dimension: $128 \rightarrow 256$	-0.030

7 Limitations and Future Work

ProtSurf is evaluated in a broad-corpus transfer setting: the RCSB pretraining source contains the PDB entries represented in SKEMPI, so the experiments do not measure generalization to structures unseen during pretraining. Although no binding-affinity labels are used for surface reconstruction, checkpoint-specific corpus provenance remains important when benchmark-derived structural examples are available. Downstream affinity prediction also depends on modeled mutant structures and on the GearBind architecture, and the present antibody evaluations cover a limited number of scaffolds. Literature comparisons may use different data splits and should therefore be interpreted as context unless methods are rerun under a common protocol. Finally, molecular-surface generation with MSMS and APBS is computationally expensive and sensitive to structure preparation. Future work should evaluate family-disjoint and post-pretraining structures, additional downstream architectures, and faster surface construction methods

8 Conclusion

We present ProtSurf, a hierarchical self-supervised framework for learning residue-aligned molecular surface representations from large-scale protein interfaces. By encoding local surface geometry and physicochemical information and contextualizing the resulting residue representations at the protein level, ProtSurf produces surface-derived embeddings that can be extracted independently of the self-supervised reconstruction objective and reused by downstream protein models.

Across model sizes from $10M$ to $650M$ parameters, we observe consistent improvements in both reconstruction quality and downstream performance on SKEMPI, while the fixed dimensionality of the final residue representation shows that these gains are not simply attributable to increasing the output embedding size. The learned representations also transfer across distinct affinity-prediction settings, including HER2 binders and SARS-CoV-2 antibody mutation ranking, when the ProtSurf encoder is kept frozen.

Overall, these results provide evidence that molecular surfaces can be treated as a scalable representation modality rather than solely as task-specific features. Residue alignment makes this surface information compatible with the residue-indexed representations used by modern protein learning systems, providing a foundation for future multimodal models that combine surface, sequence, and structural information.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and editing the manuscript text. We have not used generative AI tools to conduct experiments, generate experimental results, analyze data, or make scientific claims, and no other required disclosure tasks are applicable to this work. We have reviewed all AI-assisted text for accuracy, clarity, and consistency with the underlying research and have independently verified the scientific content, results, and claims presented in the manuscript. We take responsibility for the final content of this work, including any text produced with the aid of generative AI.

REFERENCES

Rebecca F Alford, Andrew Leaver-Fay, Jordan R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Mikhail V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 2017.

Kyle A Barlow, Shane O Conchuir, Samuel Thompson, Pooja Suresh, James E Lucas, Markus Heinonen, and Tanja Kortemme. Flex ddg: Rosetta ensemble-based estimation of changes in protein–protein binding affinity upon mutation. *The journal of physical chemistry B*, 122(21): 5389–5399, 2018.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Henry Chao, Li Chen, Paul A Craig, Gregg V Crichlow, Kenneth Dalenberg, Jose M Duarte, et al. Rcsb protein data bank (rcsb. org): delivery of experimentally-determined pdb structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic acids research*, 51(D1):D488–D508, 2023.

Huiyu Cai, Zuobai Zhang, Mingkai Wang, Bozitao Zhong, Quanxiao Li, Yuxuan Zhong, Yanling Wu, Tianlei Ying, and Jian Tang. Pretrainable geometric graph neural network for antibody affinity maturation. *Nature communications*, 15(1):7785, 2024.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M Bronstein, et al. Scaling atomistic protein binder design with generative pretraining and test-time compute. *arXiv preprint arXiv:2603.27950*, 2026.

Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, Davide Boscaini, Michael M Bronstein, and Bruno E Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

Zhangyang Gao, Cheng Tan, Jue Wang, Yufei Huang, Lirong Wu, and Stan Z Li. Foldtoken: Learning protein language via vector quantization and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 219–227, 2025.

Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. *arXiv preprint arXiv:1903.08850*, 2019.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Justina Jankauskaitė, Brian Jiménez-García, Justas Dapkūnas, Juan Fernández-Recio, and Iain H Moal. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019.

Xiaoran Jiao, Weian Mao, Wengong Jin, Peiyuan Yang, Hao Chen, and Chunhua Shen. Boltzmann-aligned inverse folding model as a predictor of mutational effects on protein-protein interactions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lzdFImKK8w.

Andy M Lau, Nicola Bordin, Shaun M Kandathil, Ian Sillitoe, Vaishali P Waman, Jude Wells, Christine A Orengo, and David T Jones. Exploring structural diversity across the protein universe with the encyclopedia of domains. *Science*, 386(6721):eadq4946, 2024.

Youhan Lee and Jaehoon Kim. Shapeprot: Top-down protein design with 3d protein shape generative model. *bioRxiv*, pp. 2023–12, 2023.

Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Liang Hong, and Pan Tan. Prosst: Protein language modeling with quantized structure and disentangled attention. *bioRxiv*, pp. 2024–04, 2024.

Shitong Luo, Yufeng Su, Zuofan Wu, Chenpeng Su, Jian Peng, and Jianzhu Ma. Rotamer density estimator is an unsupervised learner of the effect of mutations on protein–protein interaction. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=_X9Yl1K2mD.

Shitong Luo, Yufeng Su, Zuofan Wu, Chenpeng Su, Jian Peng, and Jianzhu Ma. Rotamer density estimator is an unsupervised learner of the effect of mutations on protein-protein interaction. *bioRxiv*, pp. 2023–02, 2023b.

Liheng Ma, Soumyasundar Pal, Yingxue Zhang, Philip HS Torr, and Mark Coates. Plain transformers can be powerful graph learners. *arXiv preprint arXiv:2504.12588*, 2025.

Vincent Mallet, Souhaib Attaiki, Yangyang Miao, Bruno Correia, and Maks Ovsjanikov. Atomsurf: Surface representation for learning on protein structures. *arXiv preprint arXiv:2309.16519*, 2023.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

Sungjoon Park, Soorin Yim, Dongyun Kim, Kiwoong Yoo, Doyeong Hwang, Kyungwook Lee, Jongseong Jang, and Kiyoung Kim. PreFold-dG: estimating binding affinity of protein–protein interaction from intermediate representations of protein folding model. *Bioinformatics*, 42 (Supplement_2):btag489, 2026. doi: 10.1093/bioinformatics/btag489.

Wonpyo Park, Woonggi Chang, Donggeon Lee, Juntae Kim, and Seung-won Hwang. Grpe: Relative positional encoding for graph transformer. *arXiv preprint arXiv:2201.12787*, 2022.

Sebastian Prillo and Julian Eisenschlos. Softsort: A continuous relaxation for the argsort operator. In *International Conference on Machine Learning*, pp. 7793–7802. PMLR, 2020.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

Joost Schymkowitz, Jan Borg, François Stricher, Rinaldo Nys, Frederic Rousseau, and Luis Serrano. The foldx web server: an online force field. *Nucleic Acids Research*, 33(suppl_2):W382–W388, 2005.

Sisi Shan, Shitong Luo, Ziqing Yang, Junxian Hong, Yufeng Su, Fan Ding, Lili Fu, Chenyu Li, Peng Chen, Jianzhu Ma, et al. Deep learning guided optimization of human antibody against sars-cov-2 variants with broad neutralization. *Proceedings of the National Academy of Sciences*, 119(11): e2122954119, 2022.

Amir Shanehsazzadeh, Sharrol Bachas, Matt McPartlon, George Kasun, John M Sutton, Andrea K Steiger, Richard Shuai, Christa Kohnert, Goran Rakocevic, Jahir M Gutierrez, et al. Unlocking de novo antibody design with generative artificial intelligence. *BioRxiv*, pp. 2023–01, 2023.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Patrick Soga, Zhenyu Lei, Yinhan He, Camille Bilodeau, and Jundong Li. Energy-based models for predicting mutational effects on proteins. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 2597–2608, 2025. doi: 10.1145/3711896.3736931.

Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34:25244–25255, 2021.

Zhenqiao Song, Tinglin Huang, Lei Li, and Wengong Jin. Surfpro: Functional protein design based on continuous surface. *arXiv preprint arXiv:2405.06693*, 2024.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *BioRxiv*, pp. 2023–10, 2023.

Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15272–15281, 2021.

Fangting Tao, Jinyuan Sun, Pengyue Gao, George Fu Gao, and Bian Wu. Reliable prediction of protein–protein binding affinity changes upon mutations with Pythia-PPI. *National Science Review*, 12:nwaf231, 2025. doi: 10.1093/nsr/nwaf231.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tsenkov, Sreenath Nair, Milot Mirdita, Jingi Yeo, et al. Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic acids research*, 52(D1):D368–D375, 2024.

Robin Winter, Frank Noé, and Djork-Arné Clevert. Permutation-invariant variational autoencoder for graph-level representation learning. *Advances in Neural Information Processing Systems*, 34: 9559–9573, 2021.

Fang Wu and Stan Z Li. Surface-vqmae: Vector-quantized masked auto-encoders on molecular surfaces. In *International Conference on Machine Learning*, pp. 53619–53634. PMLR, 2024.

Juhong Wu, Jiehui Sun, Tian Liang, Yongqi Zhang, Han Zhang, Tianyi Zhang, Xianmin Feng, Ping Gao, Peng Xu, and Jinyu Li. DSSA-PPI: enhancing binding affinity change prediction upon protein mutations using disentangled structure–sequence aware attention. *Chemical Science*, 17(7): 3719–3732, 2026. doi: 10.1039/D5SC08898D.

Lirong Wu, Yunfan Liu, Haitao Lin, Yufei Huang, Guojiang Zhao, Zhifeng Gao, and Stan Z. Li. A simple yet effective $\delta\delta g$ predictor is an unsupervised antibody optimizer and explainer. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, CUI Bin, Muhan Zhang, and Jure Leskovec. Vqgraph: Rethinking graph representation space for bridging gnns and mlps. In *The Twelfth International Conference on Learning Representations*, 2024.

Meng Yuan, Nicholas C Wu, Xueyong Zhu, Chang-Chun D Lee, Ray TY So, Huibin Lv, Chris KP Mok, and Ian A Wilson. A highly conserved cryptic epitope in the receptor binding domains of sars-cov-2 and sars-cov. *Science*, 368(6491):630–633, 2020.

Zuobai Zhang, Pascal Notin, Yining Huang, Aurelie C Lozano, Vijil Chenthamarakshan, Debora Marks, Payel Das, and Jian Tang. Multi-scale representation learning for protein fitness prediction. *Advances in Neural Information Processing Systems*, 37:101456–101473, 2024.

APPENDIX OVERVIEW

This appendix provides additional details supporting the main paper, including data processing, model architecture, training setup, and extended experimental results.

A PROTSURF FEATURIZATION DETAILS

A.1 LOCAL SURFACE FEATURIZATION DETAILS

Protein surfaces are computed from atomic structures using MSMS to generate a triangulated solvent-excluded surface. Surface patches are extracted following the MaSIF framework (Gainza et al., 2020) by selecting surface vertices within a geodesic radius of 9 Å around each patch center. Each patch is parameterized in local polar coordinates and discretized on a fixed grid.

For each surface patch, we compute a set of physicochemical features including shape index, distance dependent curvature, electrostatic potential (computed using APBS), hydrogen bond donor and acceptor indicators, and hydrophobicity. These features are concatenated to form a patch-level descriptor vector.

We modified the MaSIF processing code to output a vertex-to-residue mapping, enabling accurate feature computation for each surface-exposed residue. For a given residue A, we first selected all patches that are within 3Å from any of its atoms. As not all of the patches will be mapped to the residue (as per MaSIF calculation), we then categorized the patches into three groups: (1) *Core* — patches where all three vertices map to atoms from residue A, (2) *Border* — patches where at least one vertex maps to an atom from another residue, and (3) *Borrowed* — patches where no vertices map to atoms from residue A. Borrowed patches are filtered out for residue A. We assigned a label to each patch: (1) *High* — if its closest atoms include $\{C, O, N, S\}$ or all heavy atoms from its assigned residue, (2) *Medium* — if its closest atoms include $\{C, O, N, S\}$ or all heavy atoms from a neighboring residue , and (3) *Low* — for patches whose closest atoms are all hydrogen atoms. We represented each residue as a graph $G = (V, E)$, where the nodes $V = \{n_i\}_{i=1:N}$ correspond to N sampled patches following 70% from high, 20% from medium and 10% from low patches. Within each category, we adopt Farthest Point Sampling to sample the corresponding number of patches. Following Gainza et al. (2020), we set $N = 32$. We prioritized patches most closely associated with the residue, followed by those containing neighboring side-chain atoms, and lastly those consisting only of hydrogen atoms from the residue or its neighbors. When reconstructing the node features we weighted the nodes according to this classification. The above-mentioned 7-dimensional features are used as node features. The patch types are illustrated in Figure 2.

The edges $E = \{e_{ij} | j \in \mathcal{N}_i\}_{i=1:N}$ are defined where $\mathcal{N}_i = \{j | \mathrm{dist}(n_i, n_j) < 3\mathring{A}\}$ is a set of neighbors of a node n_i and $\mathrm{dist}(.,.)$ is defined as the distance between the patch centroids of nodes n_i and n_j. In addition, a virtual node has been added to the graph that is connected to every other node in the graph through its special virtual edge. This virtual node will serve a similar purpose as the [CLS] token in transformers. Since the [CLS] token has been commonly utilized to provide sentence embedding, we used the virtual node to calculate the final graph embedding and its tokenized representation. Edges are featurized according to the centroid distances: (1) *Short* — where their distance is less than 1Å, (2) *Medium* — where their distance is between 1Å and 2Å, (3) *Long* — where their distance is between 2Å and 3Å (4) *Virtual* — edge between virtual node and any other node, (5) *Self* — edge connecting each node to itself, and (6) *No* — nodes that are not connected to each other. Therefore, there are six categories of edges. Inspired by (Park et al., 2022), we have used the topological relationship $\psi(i, j)$ between nodes n_i and n_j based on their shortest path distance with the maximum cutoff *max-hop*. Formally $\psi(i, j)$ is featurized as following: (1) *Unreachable* — No connection between two nodes, (2) *shortest path distance* s — Shortest path distance value $s \in \{0, \cdots, \textit{max-hop}\}$ between nodes n_i and n_j, (3) *Far distance* — If the shortest path distance is greater than *max-hop*, and (4) *Virtual* — edge between virtual node and any other node. Therefore, there are *max-hop* + 4 categories of topological relation.

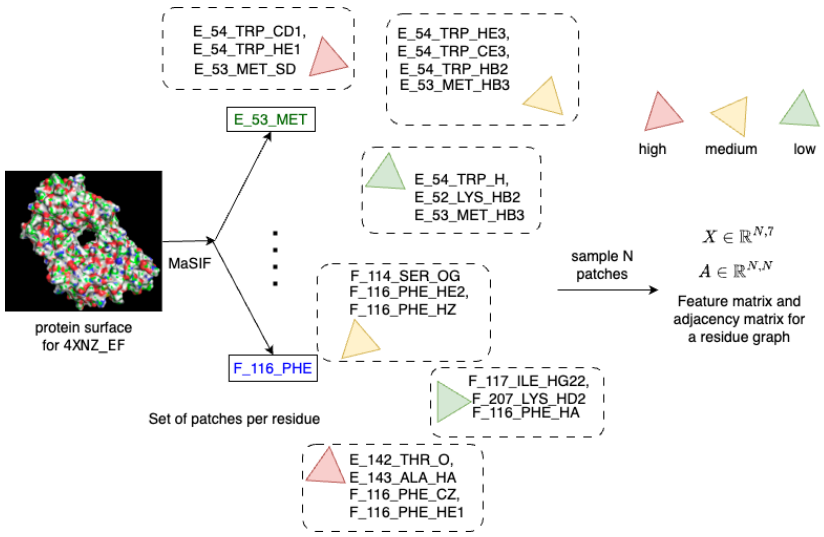

Figure 2: Creating local geometric and chemical features per residue: Given a protein structure, we first run MaSIF and get 5 features mapped to each surface exposed residue: charge, hydrophobicity, shape index, distance dependent curvature, hydrogen bond interaction. We compute 2 more features: patch centroid to C-α atom of residue and angle between C-α to patch centroid and C-α to C-β. The patches are classified as high medium or low depending on the type of core or border or borrowed atoms

A.2 Global graph per protein

Similar to creating graphs per residue, we create graphs for a given protein by using the local residue embeddings as node features and CA-CB for the surface exposed residues to denote the graph topology information. This is required to help learn the global protein geometry via a coarse approximation. We need two matrices for the global context similar to the residue graph step: shortest path distance matrix and edge type matrix which are created from the C_α atom of each surface residue. We create the shortest path distance matrix and the edge type matrix for the global modules using C-α atoms for each residue. We defined edge types based on 4Å, 8Åand 12Å. For example, for a given protein of 100 surface exposed residues there will be 100 graphs processed by the local encoder in parallel and 1 graph processed by the global module.

A.3 Local and global graph metrics

We selected the edge thresholds for both local Figure 3 and global residue graphs Figure 4 by sampling 500 random single- and multi-chain proteins from the RCSB database and computing metrics such as community sizes, number of different edges and selected parameters that led to decent community sizes and edge types.

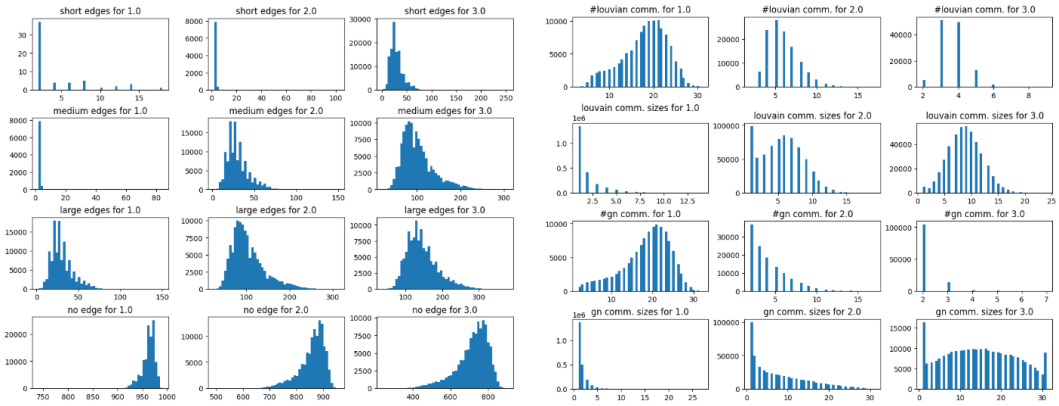

Figure 3: Graph metrics for local residue graphs

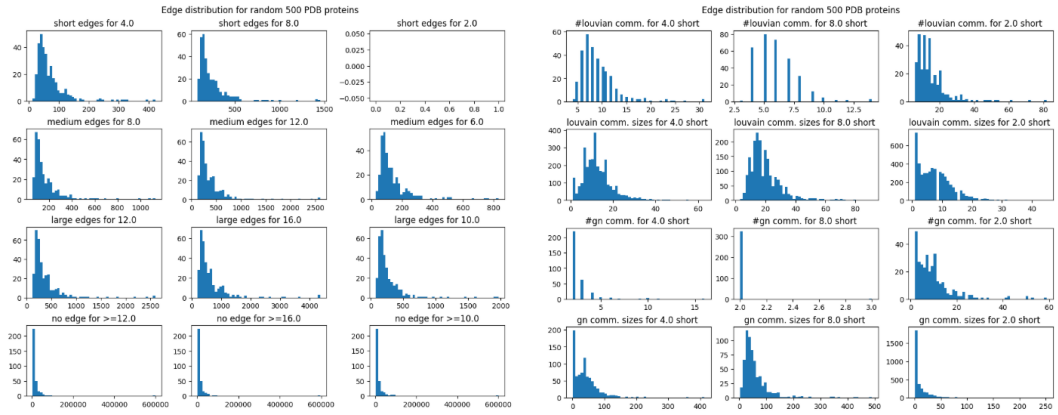

Figure 4: Graph metrics for global protein graphs

B PROTSURF COMPONENTS

B.1 LOCAL AND GLOBAL GRAPH ENCODER

We use the graph transformer with learnable relative positional encoding (Park et al., 2022), which uses (1) dot-product attention commonly used in transformers, (2) learnable topological relationship $\mathcal{P}_{\psi(i,j)} \in \mathbb{R}^{d_z}$ between nodes n_i and n_j, and (3) learnable edge relationship $\mathcal{E}_{(i,j)} \in \mathbb{R}^{d_z}$ between nodes n_i and n_j. Let us assume $x_i \in \mathbb{R}^{d_x}$ denotes the input feature of the node n_i with d_x as its dimension, and $z_i \in \mathbb{R}^{d_z}$ denotes the final output feature of transformer's layer with d_z. First, the self-attention module computes query q_i, key k_i, and value v_i with independent linear transformations $W^{\text{query}} \in \mathbb{R}^{d_x \times d_z}$, $W^{\text{key}} \in \mathbb{R}^{d_x \times d_z}$ and $W^{\text{value}} \in \mathbb{R}^{d_x \times d_z}$.

$$q_i = W^{\text{query}} x_i, k_i = W^{\text{key}} x_i \; v_i = W^{\text{value}} x_i \tag{1}$$

Second, the topological relationship between nodes n_i and n_j is calculated as:

$$a^{\text{topology}}_{(i,j)} = q_i \mathcal{P}^{\text{query}}_{\psi(i,j)} + k_i \mathcal{P}^{\text{key}}_{\psi(i,j)} \tag{2}$$

Next, the edge relationship between nodes n_i and n_j is calculated as:

$$a_{(i,j)}^{\text{edge}} = q_i \mathcal{E}_{(i,j)}^{\text{query}} + k_i \mathcal{E}_{(i,j)}^{\text{key}} \tag{3}$$

Finally, the overall attention map is computed summing these three terms. Attention here denotes full pairwise attention between the nodes adjusted by the graph features from the two additional matrices.

$$\begin{aligned} a_{(i,j)} &= \frac{q_i.k_j + a_{(i,j)}^{\text{topology}} + a_{(i,j)}^{\text{edge}}}{\sqrt{d_z}}, \\ \hat{a}_{(i,j)} &= \frac{\exp(a_{(i,j)})}{\sum_{k=1}^{N} \exp(a_{(i,k)})} \end{aligned} \tag{4}$$

The overall attention module outputs the next hidden feature by applying weighted summation on the values

$$z_i = \sum_{j=1}^{N} \hat{a}_{(i,j)} v_j \tag{5}$$

B.2 FSQ

We adopted Finite Scalar Quantization Mentzer et al. (2023) to create a protein surface codebook. FSQ creates a simple, fixed grid partition in a lower-dimensional space. Let us assume the FSQ's internal dimension is represented as d_{FSQ} and the ith dimension can have L_i different integers or *levels*. Therefore, overall *implicit* codebook size for FSQ with $\{\text{L}_1, \cdots, \text{L}_{d_{\text{FSQ}}}\}$ can be $|\mathcal{C}| = \prod_{i=1}^{d_{\text{FSQ}}} \text{L}_i$. FSQ module takes in the virtual node of a residue graph from the encoder $z_{\text{graph}} \in \mathbb{R}^{d_z}$, down-projects the graph representation down to d_{FSQ} dimension through $z_{\text{latent}} = \text{MLP}(z_{\text{graph}}) \in \mathbb{R}^{d_{\text{FSQ}}}$. Then, non-differentiable online quantization occurs for each dimension i through $z_{\text{FSQ},i} = \text{round}(\lfloor \text{L}_i/2 \rfloor \tanh(z_{\text{latent},i}))$. The quantization step binds the encoder output to L values, which is the number of dimensions of the quantizer, and then rounds to integers, leading to a quantized codebook. Since $\text{round}(.)$ function is a non-differential operation, straight-through estimator (STE) Bengio et al. (2013) can be used to propagate gradient through $\text{round_ste}(x) = x + \text{stop_gradient}(\text{round}(x) - x)$.

B.3 Residue graph decoder

Since the graph encoding from the global module is in d_{global}, we used a simple linear layer to project it back to the d_z embedding that serves the purpose of node features for the decoder denoted z_{dec}. Inspired by Winter et al. (2021), we defined sinusoidal positional embedding $\text{PE} \in \mathbb{R}^{N \times d_z}$ with the i-th node's embedding for k-th dimension as follows:

$$\text{PE}(i)_k = \begin{cases} \sin(i/10000^{2k/d_z}), & \text{if k is even} \\ \cos(i/10000^{2k/d_z}), & \text{k is odd} \end{cases} \tag{6}$$

Then we used the learned permutation matrix $\hat{\mathbf{P}}$ to reorder the positional embedding by multiplication $\text{PE}_{\text{update}} = \hat{\mathbf{P}} \times \text{PE}$. Finally, we concatenated the node features of the decoder with the updated positional encoding and passed them to the graph decoder. The graph decoder exactly follows the graph encoder with minor differences at the final project layers:

$$\begin{aligned} z_o &= g_{\text{dec}}([z_{\text{dec}} \,||\, (PE)_{\text{update}}]) \\ \hat{m}_{\text{node}} &= W_{\text{node}} z_o + b_{\text{node}} \\ \hat{m}_{\text{edge}} &= W_{\text{edge}} z_o + b_{\text{edge}} \end{aligned} \tag{7}$$

where $\hat{m}_{\text{node}}$ is used to reconstruct the initial node features m_{node} and $\hat{m}_{\text{edge}}$ is used to reconstruct the un-directed adjacency matrix $\mathbf{A}_\pi$.

B.4 PERMUTER

The permuter enables alignment between the arbitrary input node ordering and the canonical ordering learned by the decoder. The nodes in the original residue graph do not have any positional encoding and their order is arbitrary but fixed during training. Inferring this order during training allows the decoder to align the nodes which would help in the feature loss calculation. Inspired by Winter et al. (2021), we added a permuter module to reconstruct the residue graph with node features and adjacency matrix in ProtSurf. The permuter module learns to align input and output graph through *soft* alignment. Note that in ProtSurf, patches from residues form un-directed graphs with the adjacency matrix $\mathbf{A}_\pi \in \{0,1\}^{n\times n}$ for $n \in N$ in the node order $\pi \in \Pi$ with Π is the set of all permutations over V. We defined a permutation matrix $\mathbf{P} \in \mathbb{R}^{N\times N}$ that reorders nodes from order π to order π' as $\mathbf{P}_{\pi\to\pi'} = (p_{ij}) \in \{0,1\}^{n\times n}$ with $p_{ij} = 1$ if $\pi_i = \pi'_j$ and $p_{ij} = 0$ otherwise.

The inputs to the permuter are the node encodings $N \times \mathbb{R}^{d_z}$ obtained from the output of the encoder module. We discard the virtual node at this step and do not try to reconstruct it. The permuter module has to learn how the ordering of nodes in the graph generated by the decoder model will differ from a specific node order present in the input graph. During the learning process, the decoder will learn its own canonical ordering so that, given a latent code z_{latent}, it will always reconstruct a graph in that order. The permuter learns to transform/permute this canonical order to a given input node order. For each node i of the input graph, the permuter predicts a score s_i corresponding to its probability of having a low node index in the decoded graph. By sorting the input nodes indices by their assigned scores, we inferred the output node order and constructed the corresponding permutation matrix $\mathbf{P}_{\pi\to\pi'} = (p_{ij}) \in \{0,1\}^{n\times n}$ with

$$p_{ij} = \begin{cases} 1, & \text{if } j = \text{argsort}(s)_i \\ 0, & \text{else} \end{cases} \tag{8}$$

to align input and output node order. The argsort operation being non-differentiable, the continuous relaxation of the argsort operator proposed in Prillo & Eisenschlos (2020); Grover et al. (2019) has been used as follows

$$\mathbf{P} \approx \hat{\mathbf{P}} = \text{softmax}(\frac{-d(\text{sort}(s)\mathbf{1}^\top, \mathbf{1}s^\top)}{\tau}) \tag{9}$$

where the softmax operator is applied row-wise, $d(x, y)$ is the L_1-norm and $\tau \in \mathbb{R}_+$ a temperature parameter.

B.5 LOSSES

Following (Yang et al., 2024) we defined node and edge reconstruction as

$$\begin{aligned} \mathcal{L}_{\text{rec}} = \frac{1}{w_i}\sum_{i=1}^{N}(1 - \frac{m_{\text{node}}^T \hat{m}_{\text{node}}}{||m_{\text{node}}||.||\hat{m}_{\text{node}}||} \\ + ||\mathbf{A}_\pi - \sigma(\hat{m}_{\text{edge}}.\hat{m}_{\text{edge}}^T)||^2) \end{aligned} \tag{10}$$

where $\sigma(.)$ is the sigmoid function and w_i is the weight of each node (residue) normalized by the length of the protein. In order to push the *soft* permutation matrix towards a real permutation matrix (i.e. contains one 1 in every row and column), an additional penalty term was introduced to minimize the Shannon entropy both row-wise and column-wise:

$$\text{C}(\hat{\mathbf{P}}) = \sum_i \text{H}(\bar{\mathbf{p}}_i) + \sum_j \text{H}(\bar{\mathbf{p}}_j) \tag{11}$$

with Shannon entropy $\text{H}(x) = -\sum_i x_i \log(x_i)$ and normalized probabilities $\bar{\mathbf{p}}_i, = \frac{\hat{\mathbf{P}}_i}{\sum_j \hat{\mathbf{P}}_{i,j}}$.

The final loss would be:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \text{C}(\hat{\mathbf{P}}) \tag{12}$$

where λ is a hyperparameter that balances the main reconstruction loss and the additional penalty.

C PRE-TRAINING DATA

C.1 FULL COMPLEXES FROM RCSB

We constructed an RCSB Burley et al. (2023) snapshot in May 2024 containing approximately 200,000 experimentally determined, protein-containing PDB entries. The collection includes both single-chain and multi-chain entries. We split parent PDB entries into 90% pretraining and 10% pretraining-validation partitions before generating any derived examples; all examples derived from one parent entry remain in the same partition. We filter proteins that have at most 6 chains and for NMR proteins we take one copy of the protein from the PDB file.

C.2 SINGLE CHAINS FROM RCSB COMPLEXES

Approximately 130,000 entries in the processed snapshot contain at least two retained protein chains. Extracting their constituent chains produces approximately 440,000 chain-level examples. For each chain, we retain the coordinates observed in the parent multi-chain structure, remove all partner chains, and recompute the MaSIF descriptors. This operation exposes regions that are buried at the interface in the full structure. The extracted chains are treated as augmented surface views and do not use interface identities, binding-affinity measurements, or other downstream labels.

C.3 DOMAIN PROTEIN-PROTEIN INTERACTION

We used only high and medium quality structures from the AFDB database as a starting point. Structures with average Predicted Local Distance Difference Test (Avg pLDDT) greater than equal to 70% are defined as high quality and structures with Avg pLDDT between 40-70% are defined as medium quality. We obtained 113 million structures from AFDB after filtering and extracted corresponding domain information using domain information from the TED database (version 4) Lau et al. (2024). We defined two domains to be interacting if there are at least 15 residue pairs making a contact of 8Å or less for the Cβ atoms (Cα for Glycine residue). We checked whether there is an interaction or not for each pair of domains in a structure (Figure 5). After processing, the database contained about 415 million protein-protein interaction data. We further clustered the data based on 90% sequence similarity to obtain a clustered synthetic PPI dataset used for pre-training containing 74,557,665 protein-protein interactions. From this we randomly sampled 1.6M million protein-protein interactions for pre-training ProtSurf.

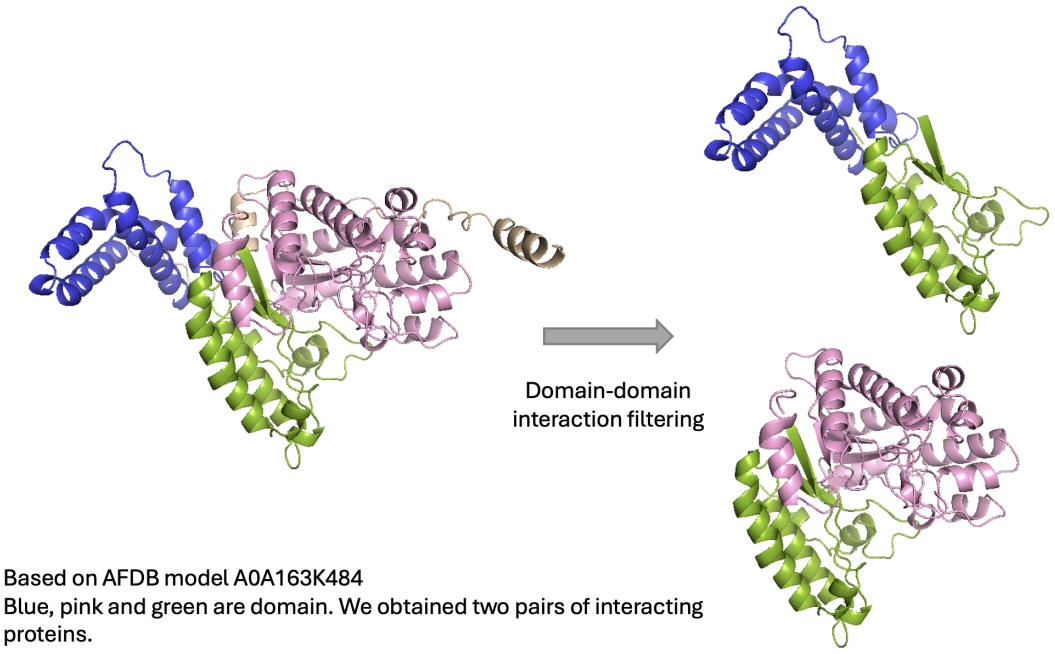

Figure 5: Curation of SynthPPI database: Cartoon representation of different domains present in the AlphaFold2 predicted structure $AF-A0A163K484-F1-model_{v4}$ with each domain shown in a different color. We split the structure into multiple domain pairs which constitute our domain interaction database. All pairs of domains are exhaustively sampled to identify interacting domains

D TRAINING DETAILS

To isolate model capacity, all five ProtSurf scales are trained for 100,000 steps on the same pretraining corpus using the same source-sampling distribution, global batch size, objective, optimizer, and learning rate. The configurations differ only in the architectural dimensions reported in Table 7. All models were trained using learning rates of $5e^{-05}$ with AdamW optimizer, weight decay 0.1, a global batch size of 512 on NVIDIA H200 GPUs for $100k$ steps. We incorporated gated attention Qiu et al. (2025), L2 attention Ma et al. (2025) and SwiGLU Shazeer (2020) into the architecture to stabilize training. We set the attention scaling hyperparameter to 10 and $alpha_{hop}$ and $alpha_{edge}$ to 0.5 in the graph transformers after HPO. We show the model size details in Table 7.

Table 7: Model architecture configurations across different scales.

	XSmall	Medium	Large	XLarge	XXLarge
#enc layers	2	8	12	16	24
#enc heads	2	8	12	16	24
enc hidden dim	256	512	768	768	1032
#global dec layers	2	4	6	8	12
#global dec heads	2	4	6	8	12
global dec hidden dim	256	512	768	768	1032
#local dec layers	4	8	12	16	24
#local dec heads	4	8	12	16	24
feedforward	1024	1024	1024	1024	2048
graph embedding dim	128	128	128	128	128
#total params (M)	10M	60M	150M	220M	650M

E RESULTS

E.1 SKEMPI RESULTS

Table 8: Evaluation of mutant effect prediction ($\Delta\Delta G$) on SKEMPI v2. Pearson and Spearman correlations are higher-is-better; MAE and RMSE are lower-is-better. Best results in **bold**, second best underlined.

Category	Method	Per-Struct.		Overall				
		Pear. ↑	Spear. ↑	Pear. ↑	Spear. ↑	RMSE ↓	MAE ↓	AUROC ↑
Energy	Rosetta	0.3284	0.2988	0.3113	0.3468	1.6173	1.1311	0.6562
	FoldX	0.3789	0.3693	0.3120	0.4071	1.9080	1.3089	0.6582
Sequence	ESM-1v	0.0073	-0.0118	0.1921	0.1572	1.9609	1.3683	0.5414
	PSSM	0.0826	0.0822	0.0159	0.0666	1.9978	1.3895	0.5260
	MSA-T	0.1031	0.0868	0.1173	0.1313	1.9835	1.3816	0.5768
	Tranception	0.1348	0.1236	0.1141	0.1402	2.0382	1.3883	0.5885
Supervised	DDGPred	0.3750	0.3407	0.6580	0.4687	1.4998	1.0821	0.6992
	End2End	0.3873	0.3587	0.6373	0.4882	1.6198	1.1761	0.7172
Pretrain	ESM-IF	0.2241	0.2019	0.3194	0.2806	1.8860	1.2857	0.5899
	B-factor	0.2042	0.1686	0.2390	0.2625	2.0411	1.4402	0.6044
	MIF-Net	0.3965	0.3509	0.6523	0.5134	1.5932	1.1469	0.7329
	RDE-Net.	0.4422	0.3914	0.6386	0.5340	1.5904	1.1307	0.7380
	PPIFormer	0.4281	0.3995	0.6450	0.5304	1.6420	1.1186	0.7380
	Surf-VQMAE	0.4694	0.4324	0.6482	0.5611	1.5876	1.1271	0.7469
	Prompt-DDG	0.4369	0.3904	0.6661	0.5626	1.5415	1.0989	0.7369
	AtomSurf	0.2321	0.2192	0.3593	0.3168	1.8210	1.5091	0.6019
	GearBind+PT	0.4895	0.4340	0.6760	0.5250	1.6110	1.1150	0.7510
	BA-DDG	0.5266	0.5040	0.6469	0.6034	1.6650	1.1942	0.7533
	EBM-DDG	0.5472	0.5076	0.7140	**0.6526**	1.4436	**1.0102**	**0.7866**
	ProtSurf (650M)	**0.5987**	**0.5260**	**0.7612**	0.6240	**1.4131**	1.0119	0.7701

E.2 RESULTS WITH STANDARD DEVIATION

Results with standard deviation. We repeated experiments across 5 random seeds and 3 data splits, resulting in 15 independent runs. The low variance across seeds demonstrates that ProtSurf exhibits stable performance and that improvements with model scaling are statistically consistent. See Table 9.

Table 9: Model performance comparison on SKEMPI v2, reported as mean ± standard deviation across 15 runs (5 random seeds × 3 data splits). Central values match Table 5.

Model	**PearsonR**	**SpearmanR**	**MAE**	**RMSE**
10M	0.6856 ± 0.0023	0.5450 ± 0.0013	1.1853 ± 0.0013	1.5525 ± 0.0029
60M	0.7033 ± 0.0012	0.5686 ± 0.0043	1.1402 ± 0.0025	1.4969 ± 0.0042
150M	0.7344 ± 0.0015	0.5931 ± 0.0048	1.0910 ± 0.0047	1.4907 ± 0.0050
220M	0.7419 ± 0.0078	0.6078 ± 0.0059	1.0461 ± 0.0032	1.4727 ± 0.0041
650M	0.7612 ± 0.0039	0.6240 ± 0.0024	1.0119 ± 0.0055	1.4131 ± 0.0070

E.3 HER2 SCALING RESULTS

Table 10: Correlation on HER2 $\Delta\Delta G$ prediction. Pearson and Spearman correlations are higher-is-better. Best results in **bold**.

Method	Pearson ↑	Spearman ↑
FoldX	0.246	0.329
RDE-Net.	0.465	0.481
Bind-ddG	0.387	0.388
BA-DDG	0.326	0.362
AtomSurf	0.100	0.071
GearBind+Pretrained	0.515	0.517
ProtSurf (10M)	0.516	0.517
ProtSurf (60M)	0.519	0.521
ProtSurf (150M)	0.528	0.529
ProtSurf (220M)	<u>0.529</u>	<u>0.531</u>
ProtSurf (650M)	**0.534**	**0.536**

F ABLATIONS

F.1 HIERARCHICAL GLOBAL MODELING IS CRITICAL, AND PLACEMENT MATTERS

To evaluate the role of global context, we compare architectural variants on a $60M$ ProtSurf trained on $200k$ RCSB proteins that introduce the global module at different stages, along with a purely local baseline . Applying global context before the reconstruction stage (Local Encoder–Global Module) yields the strongest improvement over the local baseline (Pearson $0.600 \rightarrow 0.641$). In contrast, introducing global modeling earlier (Global Module–Local Decoder) or throughout the network provides smaller gains (Pearson 0.616 and 0.619, respectively). These results suggest that preserving local surface structure prior to global aggregation is important, and that introducing global context before the quantization stage can degrade useful long-range information. This validates the hierarchical local-to-global design of ProtSurf.

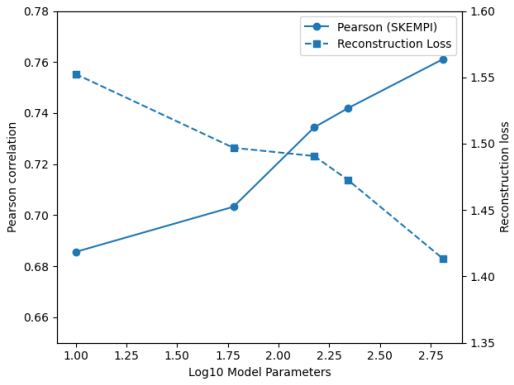

Figure 6: As model capacity increases, pretraining reconstruction loss decreases (dotted line) while downstream performance on SKEMPI improves (solid line) indicating that improved surface representation learning transfers to binding affinity prediction.

Local Surface Embedding with Global-Aware Reconstruction Virtual node embeddings are first quantized using Finite Scalar Quantization (FSQ) to produce quantized tokens and their corresponding quantized embeddings. These quantized embeddings are then processed by a global module operating on a single protein-level graph defined by C_α–C_α SPD and edge types. Padding and dynamic masking are applied to support variable protein lengths. The globally updated quantized embeddings are merged with local node embeddings via a permuter and passed to the local decoder Figure 7

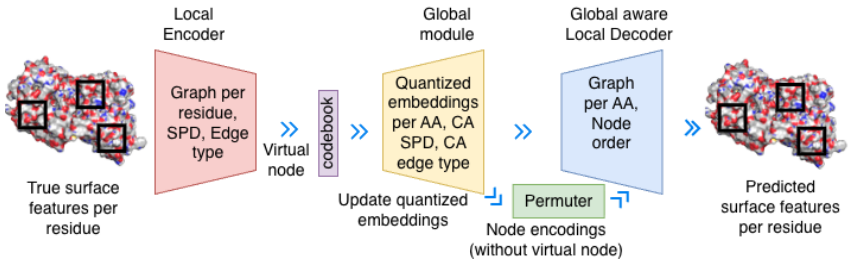

Figure 7: Local Encoder Global Module

Global Surface Embedding with Local Reconstruction Virtual node embeddings are first updated by the global module at the protein level and subsequently quantized using FSQ. The resulting globally aware quantized embeddings are combined with local node embeddings and passed to the decoder, yielding globally informed but locally reconstructed surface features Figure 8.

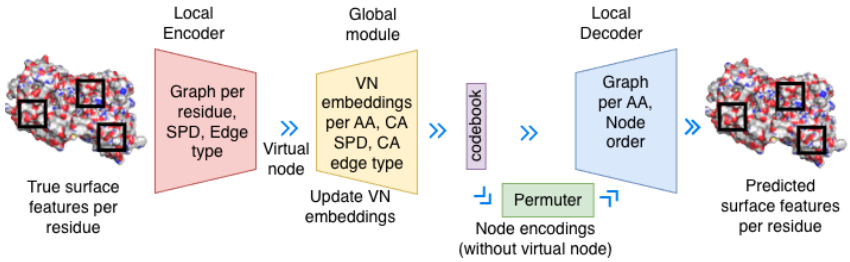

Figure 8: Global Module Local Decoder

Global Surface Embedding with Global-Aware Reconstruction This hybrid variant applies global contextualization both before and after quantization. Virtual node embeddings are globally updated, quantized, and then refined by a second global module. The resulting embeddings are permuted and decoded to produce surface reconstructions Figure 9.

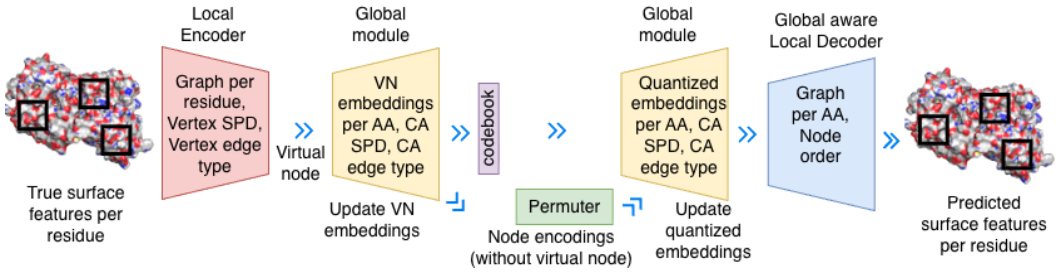

Figure 9: Global Encoder Global Module

Local Surface Embedding This variant gets rid of the global module altogether. Virtual node embeddings are quantized, and the resulting embeddings are permuted and decoded to produce surface reconstructions Figure 10.

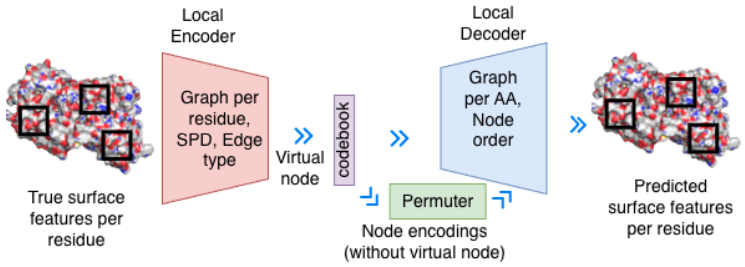

Figure 10: Local Encoder Local Decoder

F.2 Discrete surface tokenization trades compression for structure

To understand the role of the quantization bottleneck, we remove the finite-scale quantization (FSQ) layer and train a 60M model on the same training corpus. Removing the quantizer changes Pearson correlation on SKEMPI from 0.7033 to 0.7094, a difference within the seed-to-seed variation observed on this benchmark. We therefore do not attribute a downstream accuracy benefit to quantization. Its role in ProtSurf is to yield a finite per-residue codebook, which provides a natural representational form for using the surface as a token stream in residue-indexed multimodal protein models, and which additionally permits codebook-level analysis of recurring surface patterns.

F.3 Surface-specific geometric features provide important inductive bias

We assess the contribution of the two additional geometric features introduced beyond standard MaSIF inputs by training a $60M$ model on the same training corpus but using only the original five features. Removing these features leads to a substantial drop in performance (Pearson $0.7033 \rightarrow 0.6412$). We refer to these as 'anchor features'; their removal substantially degrades performance, indicating that residue-centric geometric signals provide an important inductive bias for surface representation learning.

F.4 Graph reconstruction beyond node features improves representation learning

We evaluate the role of the permutation module by removing it and training a $60M$ model on the same training corpus that reconstructs only node features without adjacency structure. This results in a performance drop on SKEMPI (Pearson $0.7033 \rightarrow 0.6813$), demonstrating that learning surface connectivity through adjacency reconstruction provides a stronger supervision signal. These results highlight the importance of modeling both node features and graph structure for effective surface representation learning.

F.5 Model capacity improves performance but depends on data

Scaling model capacity from 10M to 60M parameters yields consistent improvements in both reconstruction loss (from $0.0545 \rightarrow 0.053$ on $200k$ proteins and $0.0517 \rightarrow 0.0509$ on $600k$ proteins) and downstream performance (Pearson $0.685 \rightarrow 0.703$ on SKEMPI). However, these gains are closely tied to the availability of sufficient and diverse pretraining data, indicating that model capacity alone is not sufficient without appropriate data scaling.

F.6 Latent embedding dimension affects representation quality

We evaluate the effect of varying the latent embedding dimension while keeping model size fixed. Using a 150M model, both lower (64) and higher (256) dimensions underperform the default 128-dimensional setting (Pearson 0.719 and 0.704 vs 0.734), indicating that a fixed intermediate representation provides a better balance between expressivity and generalization.