Effective Protein-Protein Interaction Exploration with PPIretrieval

Chenqing Hua McGill & Mila

Connor Coley MIT

Guy Wolf UdeM & Mila

Doina Precup McGill & Mila

Shuangjia Zheng SJTU

Abstract

Protein-protein interactions (PPIs) are crucial in regulating numerous cellular functions, including signal transduction, transportation, and immune defense. As the accuracy of multi-chain protein complex structure prediction improves, the challenge has shifted towards effectively navigating the vast complex universe to identify potential PPIs. Herein, we propose PPIretrieval, a retrieval model for protein-protein interaction exploration, which leverages existing PPI data to effectively search for potential PPIs in an embedding space, capturing rich geometric and chemical information of protein surfaces. When provided with an unseen query protein with its associated binding site, PPIretrieval effectively identifies a potential binding partner along with its corresponding binding site in an embedding space, facilitating the formation of protein-protein complexes. Our codes are available on `https://anonymous.4open.science/r/ppi_search-9E39`.

1 Introduction

Proteins are the building blocks of life, engaged in a myriad of interactions [39]. Understanding how proteins interact with each other is fundamental to unraveling the intricate machinery of biological systems [7, 39]. Therefore, the ability to predict and analyze protein-protein interactions (PPIs) not only improves our understanding of cellular functions but also plays a pivotal role in drug discovery [8, 30, 1]. Current methods try to analyze, understand, and design PPIs [34, 9, 28, 10], but the results are often constrained by the complexity of protein interactions and the limited understanding of the underlying mechanisms. Despite the progress, there is a pressing need for more effective strategies aimed at designing new protein binders. This objective is crucial for advancing therapeutic interventions and understanding the protein interactions of various biological processes.

Geometric deep learning emerges as a potent strategy for representing and learning about proteins, focusing on structured, non-Euclidean data like graphs and meshes [29, 24, 22, 23, 14, 13]. In this context, proteins can be effectively modeled as graphs, where nodes correspond to individual atoms or residues, and edges represent interactions between them [12, 45]. In addition, protein structures can be represented as point clouds or meshes, wherein each point or vertex corresponds to an atom or a residue [9, 35]. Indeed, representing proteins as graphs or point clouds offers a valuable approach for gaining insights into and learning the fundamental geometric and chemical mechanisms governing PPIs [28, 10, 38]. This representation allows for a more comprehensive exploration of the intricate relationships and structural features within protein structures [36, 15].

Instead of directly generating PPIs, given the limited understanding of their underlying biological mechanisms, we draw inspiration from a previously developed model known as Foldseek [37]. This model aligns the structure of a query protein against a database by representing its tertiary structure in an embedding space. Rather than attempting to build a generative model that may inaccurately interpret the complex mechanisms of PPIs, we can utilize existing PPI data to develop a retrieval model for PPIs in an embedding space.

Therefore, we introduce a protein-protein-interaction retrieval model, namely **PPIretrieval**, which leverages existing PPI data to search for potential PPIs in an embedding space with rich geometric and chemical information. In our approach, PPIretrieval learns surface representations of PPI complexes, which are initially represented as two point clouds along with their corresponding binding interfaces. Then, the embedded surface representations with information about binding partners are stored in our

AI for New Drug Modalities at NeurIPS 2024.

database for further comparison. When provided with an unseen query protein, PPIretrieval learns its surface representation and retrieves the most similar surface representation along with its known binding partner in our database. PPIretrieval outputs the binding partner along with predicted binding interface for the query protein, enabling the exploration of potential PPIs.

Why a Retrieval Model? The retrieval model is designed to leverage the wealth of existing PPI data. Unlike generative models [40, 43, 4, 25], our approach focuses on retrieving potential binders directly from a curated surface database. When presented with a query protein P, if it shares sequential or structural similarities with proteins in our database, the model can identify and retrieve binders that are structurally complementary to P. This method addresses the limitations of generative models, which may inaccurately predict binder size or overlook shape complementary.

Addressing the Retrieval Challenge Retrieving a potential binding partner poses a dual challenge: the binding sites must not only exhibit similar representations but also be shape complementary. Our model incorporates principles from shape correspondence studies to learn these complementary interactions and emulate the lock-and-key mechanism. Furthermore, we conduct experiments comparing our heat message-passing (heat-MP) surface encoder with different encoders, *e.g.,* MPNN [11], GIN [42], implicit representations [32], which empirically validate our model's efficacy in Sec. 7.2.

Generalization Assurance PPIretrieval is not designed to memorize specific PPIs or predict particular binding sites. Instead, it is trained to optimize parameters that capture the universal *lock-and-key* structures inherent to receptor-ligand interactions (Sec. 4). Through fine-grained surface-level learning, which encapsulates chemical and geometric features, our model is optimized to capture the fundamental complementary structures between receptors and ligands (Sec. 7.1 cross-data validation).

Potential Application The potential application of our proposed method lies in its ability to facilitate the discovery and design of novel protein binders by accurately retrieving entire proteins that are likely to interact with a given target protein (Sec. 5). Unlike traditional PPI classification tasks that typically predict whether a pair of proteins will interact, our method focuses on identifying potential binding partners from a vast database of known interactions. To clarify, PPIretrieval is a ligand-specific method, while dMaSIF [35] and MaSIF [9] are not ligand-specific. *One promising approach is to jointly use use dMaSIF/MaSIF with PPIretrieval. For a new folded protein structure, one may use dMaSIF/MaSIF (non ligand-specific method) to determine the active site of it, followed by PPIretrieval (ligand-specific method) to identify top-n binders.*

2 Preliminaries

In PPIretrieval, the objective is to identify a binding partner B with its corresponding binding interface for an unseen query protein P with a known binding interface using surface representations.

Protein Surface Representation A protein is a chain of residues $P = \{\mathbf{a}_1, ..., \mathbf{a}_{N_P}\} \in [0,1]^{N_P \times 20}$, consisting of N_P residues in Euclidean space $\upsilon_P = \{r_1, ..., r_N\} \in \mathbb{R}^{N_P \times 3}$. The binding interface $\mathbf{Y}_P^{\text{res}} \in \{0,1\}^{N_P \times 1}$ of protein P denotes a region where the protein is poised to interact with another protein, forming a complex. The protein P can be characterized by a set of surface points $\{x_1, ..., x_{M_P}\} \in \mathbb{R}^{M_P \times 3}$ with unit normals $\{n_1, ..., n_{M_P}\} \in \mathbb{R}^{M_P \times 3}$ located on its surface. In this context, we employ the surface representation to learn the protein, which captures its structural characteristics and potential interactions with other proteins.

LB Operator on Protein Surface The Laplace-Beltrami (LB) operator is known for its smooth operation on compact Riemannian manifolds [19, 38]. When applied to the protein surface, which can be considered as a 2D Riemannian manifold $\mathcal{M}$ with a Laplace-Beltrami (LB) operator $\Delta_{\mathcal{M}}$, the LB operator operates smoothly. The LB operator Δ has an eigendecomposition $\Delta\phi_i = \lambda_i\phi_i,\ 0 \leq \lambda_1 \leq \lambda_2 \leq ...$, and the set $\{\phi_1, \phi_2, ...\}$ forms orthonormal basis for the space of functions defined on $\mathcal{M}$. Any function on the surface can be expressed as a linear combination of these eigenfunctions,

$$g = \sum_i a_i\phi_i,\ a_i = \langle g, \phi_i \rangle_{\mathcal{M}}. \tag{1}$$

Here, g is a linear combination of basis functions, with the scalars a_i determined by the inner product defined on the surface.

LB Operator-Induced Message-passing Consider $g(x,t)$ as the measure of heat at point x on the surface at time t. The heat operation is a message-passing process (heat-MP), smoothly propagating information from hot regions to cool regions. The change in heat over time is described by the LB operator $\frac{\partial g}{\partial t} = \Delta g$ [19]. After time t, the heat distribution is equivariant to

$$g(x,t) = \exp\left(H_t\right)g,\ H_t = -\Delta t = -\lambda t. \tag{2}$$

where H_t denotes the heat operator at time t. Following Eq. 1, we can define a function propagation operator $\mathcal{F}$ on the heat distribution over the surface [38],

$$\mathcal{F}g(x,t) = \sum_i F_i \exp(-\lambda_i t)\langle g, \phi_i\rangle_{\mathcal{M}}\phi_i, \tag{3}$$

where $\{F_i\}$ are frequency filters, $\{\lambda_i\}$ are eigenvalues, $\{\phi_i\}$ are eigenfunctions, and t is a time parameter for the operator. [38] propose to use a linear Gaussian filter to learn frequency, $F_i = F_{(\mu,\sigma)}(\lambda_i) = \exp(-\frac{(\lambda_i-\mu)^2}{\sigma^2})$, where μ, σ^2 are learned parameters for mean and variance. This LB operator-induced message-passing is employed for smooth and effective information flow on the (compact) protein surface, as discussed in Sec. 3.1.

3 PPIretrieval

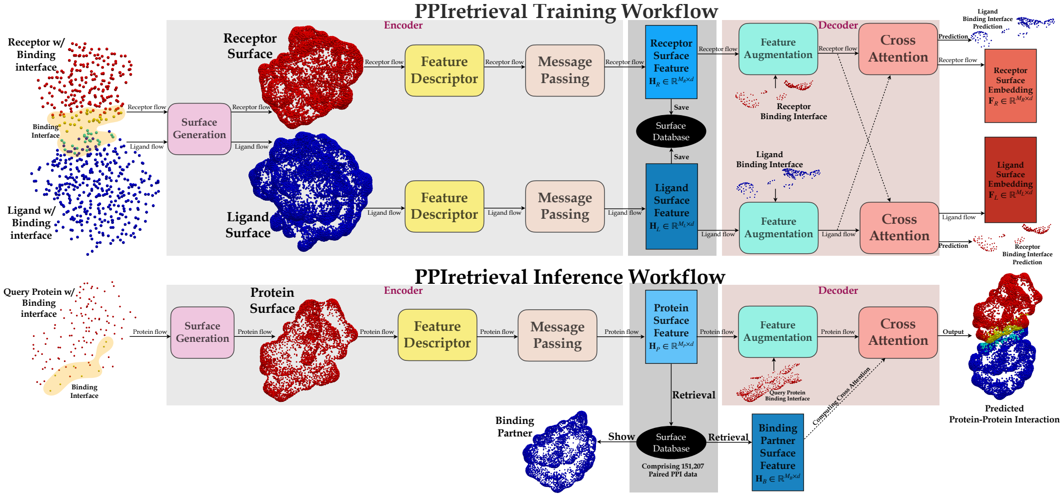

PPIretrieval Training Workflow
Encoder
Receptor w/ Binding interface
Binding Interface
Ligand w/ Binding interface
Receptor flow
Ligand flow
Surface Generation
Receptor Surface
Ligand Surface
Feature Descriptor
Message Passing
Receptor Surface Feature $\mathbf{H}_R \in \mathbb{R}^{M_R \times d}$
Save
Surface Database
Ligand Surface Feature $\mathbf{H}_L \in \mathbb{R}^{M_L \times d}$
Decoder
Feature Augmentation
Receptor Binding Interface
Ligand Binding Interface
Cross Attention
Prediction
Ligand Binding Interface Prediction
Receptor Surface Embedding $\mathbf{F}_R \in \mathbb{R}^{M_R \times d}$
Ligand Surface Embedding $\mathbf{F}_L \in \mathbb{R}^{M_L \times d}$
Receptor Binding Interface Prediction
PPIretrieval Inference Workflow
Query Protein w/ Binding interface
Binding Interface
Protein flow
Protein Surface
Protein Surface Feature $\mathbf{H}_P \in \mathbb{R}^{M_P \times d}$
Query Protein Binding Interface
Output
Predicted Protein-Protein Interaction
Retrieval
Show
Binding Partner
Comprising 151,207 Paired PPI data
Binding Partner Surface Feature $\mathbf{H}_B \in \mathbb{R}^{M_B \times d}$
Computing Cross Attention

Figure 1: An overview of PPIretrieval pipeline, demonstrating the training and inference workflows. **During training**, PPIretrieval processes a PPI complex. These embeddings with their binding partnership are stored in our database. **During inference**, PPIretrieval takes a protein P with its corresponding binding interface to the `encoder`. Then, it identifies a binding partner B from our database. The `decoder` takes them to predict the PPI complex. Details can be found in App. A.

The overview of PPIretrieval is demonstrated in Fig. 1. PPIretrieval follows a specific design process. Starting with a query protein P comprising N_P residues in Euclidean space $\upsilon_P \in \mathbb{R}^{N_P \times 3}$, along with its corresponding binding interface $\mathbf{Y}_P^{\text{res}}$, we first encode the protein into a surface representation $\mathbf{H}_P \in \mathbb{R}^{M_P \times d}$. The next step involves retrieving the most similar surface feature $\mathbf{H}_A \in \mathbb{R}^{M_A \times d}$ in the database. Once identified, we locate a binding partner protein B that binds to A, with a surface feature $\mathbf{H}_B \in \mathbb{R}^{M_B \times d}$. Then, the model decodes the pair of protein surface features, $\mathbf{H}_P, \mathbf{H}_B$, utilizing the known binding interface $\mathbf{Y}_P^{\text{res}}$. The final prediction involves estimating the binding interface $\hat{\mathbf{Y}}_B^{\text{res}}$ for protein B. In summary, the model outputs the protein B that is most likely to bind to the input protein P, accompanied by the predicted binding interface $\hat{\mathbf{Y}}_B^{\text{res}}$.

3.1 Surface Representation Encoder

The surface encoder network aims to encode an input protein P into a surface representation $\mathbf{H}_P \in \mathbb{R}^{M_P \times d}$, where M_P denotes the number of surface points representing P. This representation captures the propagated chemical and geometric information. It can be stored in our database for subsequent retrieval and comparison purposes.

Protein Surface Preparation The input consists of one-hot encoded residue types for protein $P \in [0,1]^{N_P \times 20}$ in Euclidean space $\upsilon_P \in \mathbb{R}^{N_P \times 3}$, along with the corresponding binding interface $\mathbf{Y}_P^{\text{res}} \in \{0,1\}^{N_P \times 1}$. We apply dMaSIF [35] to generate a set of oriented surface points $\{x_1, ..., x_{M_P}\} \in \mathbb{R}^{M_P \times 3}$ with unit normals $\{n_1, ..., n_{M_P}\} \in \mathbb{R}^{M_P \times 3}$ to approximate a smooth manifold representing the surface of protein P.

Upon it, we define the LB operator Δ_P on the surface for heat-MP. We compute the first k eigenfunctions of Δ_P stacked in matrix $\Phi_P \in \mathbb{R}^{M \times k}$ with their corresponding eigenvalues $\{\lambda_i\}_{i=1}^{k}$ ($k = 100$), then calculate the Moore-Penrose pseudo-inverse of the eigenfunction matrix, $\Phi_P^+ \in \mathbb{R}^{k \times M}$. Heat-MP allows effective information flow between surface points, crucial for message propagation [38].

Geometric & Chemical Descriptors Following the computation of surface points to represent the input protein, we describe the local geometric features of the surface. We approximate the per-point mean curvature, Gaussian curvature as detailed in [5], and compute the Heat Kernel Signatures as described in [33]. The geometric features processed by a MLP, $\mathbf{F}_{\text{Geom}} \leftarrow \texttt{MLP}([\mathbf{F}_{\text{Mean}}, \mathbf{F}_{\text{Gauss}}, \mathbf{F}_{\text{HKS}}]) \in \mathbb{R}^{M_P \times d_G}$, capture the local geometric environment for each surface point.

We proceed to compute chemical features for surface points based on the one-hot encoded residue types and the binding interface. This is computed through a *multi-level message-passing process* by first propagating information between residues, then propagating information from residues to surface points via radius graphs, resulting in per-point features $\mathbf{F}_{\text{Chem}} = \texttt{Chem-Descriptor}(P, \mathbf{Y}_P^{\text{res}}, \upsilon_P) \in \mathbb{R}^{M_P \times d_C}$. Implementation details can be found in App. A.1.

Finally, we use a MLP to combine them, $\mathbf{F}_{\text{Surf}} \leftarrow \texttt{MLP}([\mathbf{F}_{\text{Geom}}, \mathbf{F}_{\text{Chem}}]) \in \mathbb{R}^{M_P \times d}$. For each surface point, these features capture the local geometric and chemical environment, along with binding interface information. These can be effectively used in the heat-MP for facilitating information flow on the approximated protein surface.

Message-passing on Protein Surface Then, we perform heat message-passing on protein surfaces, resulting in surface-level embeddings $\mathbf{H}_P = \texttt{Heat-MP}(\mathbf{F}_{\text{Surf}}, \Phi_P, \Phi_P^+, \lambda) \in \mathbb{R}^{M \times d}$. This operation treats the protein surface as a compact object [38]. Implementation details can be found in App. A.1.

Training Stage During training, the encoder network is fed with the paired receptor and ligand proteins R, L, along with their corresponding binding interfaces $\mathbf{Y}_R^{\text{res}}, \mathbf{Y}_L^{\text{res}}$. It then processes these inputs to generate distinct surface features for each protein, $\mathbf{H}_R = \texttt{Encoder}(R, \mathbf{Y}_R^{\text{res}}) \in \mathbb{R}^{M_R \times d}$, $\mathbf{H}_L = \texttt{Encoder}(L, \mathbf{Y}_L^{\text{res}}) \in \mathbb{R}^{M_L \times d}$.

Surface-level Binding Interface We construct the surface-level binding interface $\mathbf{Y}_P^{\text{surf}} \in \{0,1\}^{M_P \times 1}$ for surface points based on the residue-level binding interface $\mathbf{Y}_P^{\text{res}} \in \{0,1\}^{N_P \times 1}$ for protein P. For each residue in P, we define a region with a predefined radius of $r = 10\text{Å}$. All surface points falling within this region are then labeled as part of the surface binding interface.

3.2 Interactive Decoder

The decoder network operates by taking surface features as input, allowing interaction between two proteins, and ultimately predicting a binding interface.

We assume that input data comprises the receptor R and its corresponding binding interface $\mathbf{Y}_R^{\text{res}}$. The encoder network generates a surface feature $\mathbf{H}_R \in \mathbb{R}^{M_R \times d}$. Simultaneously, we compute the surface-level binding interface $\mathbf{Y}_R^{\text{surf}} \in \{0,1\}^{M_R \times 1}$. Then, the model identifies a binding partner L and obtains its own surface feature $\mathbf{H}_L \in \mathbb{R}^{M_L \times d}$. The objective is to predict the binding interface for ligand L, expressed as $p(\hat{\mathbf{Y}}_L^{\text{res}} | \mathbf{H}_R, \mathbf{H}_L, \mathbf{Y}_R^{\text{surf}}) \in [0,1]^{N_L \times 1}$.

Cross-Attention Given the surface-level features $\mathbf{H}_R, \mathbf{H}_L$ of the receptor and ligand along with their respective geometries x_R, x_L, we employ a cross-attention module to compute their interactions, $\mathbf{F}_R, \mathbf{F}_L = \texttt{Cross-Attn}(\mathbf{H}_R, x_R, \mathbf{H}_L, x_L) \in \mathbb{R}^{M_R \times d}, \mathbb{R}^{M_L \times d}$. Implementation details can be found in App. A.2.

Binding Interface Once we obtain the propagated surface features for the ligand $\mathbf{F}_L \in \mathbb{R}^{M_L \times d}$, we employ a MLP with sigmoid function directly on these features, predicting for the binding interface, $\hat{\mathbf{Y}}_L^{\text{surf}} \leftarrow \sigma(\texttt{MLP}(\mathbf{F}_L)) \in [0,1]^{M_L \times 1}$. Thus, the prediction of surface-level binding interface for ligand L is solely conditioned by the surface features of both the receptor and the ligand, along with the binding interface information of the receptor.

Surface Point to Residue We compute the residue-level binding interface $\hat{\mathbf{Y}}_L^{\text{res}} \in \{0,1\}^{N_L \times 1}$ from $\hat{\mathbf{Y}}_L^{\text{surf}}$, and embedding $\hat{\mathbf{F}}_L \in \mathbb{R}^{N_L \times d}$ from $\mathbf{F}_L$. For each residue i in L, we define a region with a fixed radius of $r = 10\text{Å}$ and collect a set of surface points within this region, each with a binding interface $\hat{\mathbf{y}}_j^{\text{surf}}$ and embedding $\mathbf{f}_j$. The residue i is considered part of the binding interface if the majority of surface points in the region are labeled as part of the binding interface, i.e., $\hat{\mathbf{y}}_i^{\text{res}} = 1$ if

$\texttt{Mean}(\sum_j \hat{\mathbf{y}}_j^{\text{surf}}) > 0.5$; otherwise $\hat{\mathbf{y}}_i^{\text{res}} = 0$. And the residue-level embedding is obtained using the same logic, where $\hat{\mathbf{f}}_i = \texttt{Mean}(\sum_j \mathbf{f}_j)$.

Training Stage During training, each PPI sample is treated as two training instances. The model first takes the receptor R and its associated binding interface $\mathbf{Y}_R^{\text{surf}}$ as input, predicting the ligand's binding interface; then the model takes the ligand L and its corresponding binding interface $\mathbf{Y}_L^{\text{surf}}$ as input, predicting the receptor's binding interface, $\hat{\mathbf{Y}}_L^{\text{res}}, \hat{\mathbf{F}}_L = \texttt{Decoder}(\mathbf{H}_R, \mathbf{H}_L, \mathbf{Y}_R^{\text{surf}}),\ \ \hat{\mathbf{Y}}_R^{\text{res}}, \hat{\mathbf{F}}_R = \texttt{Decoder}(\mathbf{H}_R, \mathbf{H}_L, \mathbf{Y}_L^{\text{surf}})$.

4 Training Objective

Consider $\hat{\mathbf{F}}_R \in \mathbb{R}^{N_R \times d}, \hat{\mathbf{F}}_L \in \mathbb{R}^{N_L \times d}$ as the propagated surface features derived from our interactive decoder model for the receptor and ligand proteins. Additionally, let $\hat{\mathbf{Y}}_R^{\text{res}} \in [0,1]^{N_R \times 1}, \hat{\mathbf{Y}}_L^{\text{res}} \in [0,1]^{N_L \times 1}$ denote the predicted binding interface for the receptor and ligand protein, respectively. The optimization aims to utilize the *lock-and-key* structure within a PPI complex.

4.1 Lock-and-Key Optimization

In the modeling, we assume an entirely rigid protein structure. Within PPIs, a *lock-and-key* structure is established between the rigid proteins, where their structures exhibit complementary representations [27]. To utilize the structure, we optimize the model to learn the *lock-and-key* counterpart and pairwise matching between residue features $\hat{\mathbf{F}}_R, \hat{\mathbf{F}}_L$, inspired from shape correspondences [16, 21].

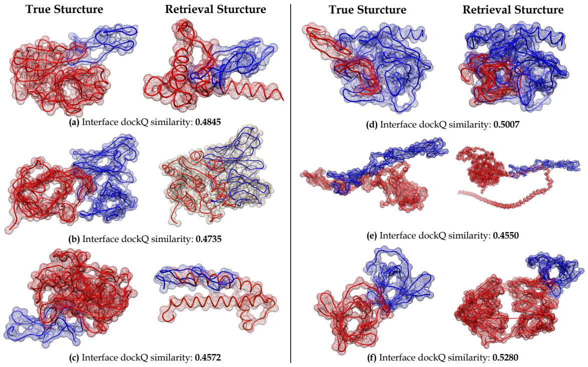
Affinity Metric Given $\hat{\mathbf{F}}_R, \hat{\mathbf{F}}_L$, we compute the global affinity matrix $\mathbf{A} \in \mathbb{R}^{N_R \times N_L}$ and its corresponding doubly-stochastic matrix $\hat{\mathbf{X}} \in [0,1]^{N_R \times N_L}$ as follows, $\mathbf{A} = \exp{(\hat{\mathbf{F}}_R^T W \hat{\mathbf{F}}_L / \tau_{\mathbf{A}})}, \hat{\mathbf{X}} = \texttt{sinkhorn}(\mathbf{A})$, where $W \in \mathbb{R}^{d \times d}$ consists of learnable affinity weights and $\tau_{\mathbf{A}}$ denotes the temperature hyperparameter. $\hat{\mathbf{X}}$ is a doubly-stochastic matrix computed by the differentiable sinkhorn layer [6], where $\hat{\mathbf{X}}_{ij}$ measures the soft-matching score between surface features $\hat{\mathbf{f}}_i \in \hat{\mathbf{F}}_R, \hat{\mathbf{f}}_j \in \hat{\mathbf{F}}_L$.

Lock-and-Key We construct a ground-truth matching matrix $\mathbf{X} \in \{0,1\}^{N_R \times N_L}$ to optimize the soft-matching score between two proteins as follows, $\mathbf{x}_{ij} = 1$ if $d_{ij} \leq d_{\text{cut}}, \mathbf{x}_{ij} = 0$ otherwise. Here $\mathbf{x}_{ij} = 1$ only if the pairwise Euclidean distance between two residues $i \in R, j \in L$ is within a cutoff distance $d_{\text{cut}} = 10\text{Å}$, implying that they are close enough to interact; otherwise $\mathbf{x}_{ij} = 0$.

To optimize PPIretrieval from the *lock-and-key* perspective, we enforce the soft-matching score to closely resemble the ground-truth matching matrix, $\mathcal{L}_{\text{match}} = \texttt{BCE}(\hat{\mathbf{X}}, \mathbf{X})$. The matching loss serves the dual purpose of encouraging a close alignment between the soft-matching scores and the ground truth, as well as providing global rigidity guidance by ensuring each residue is matched with its complementary part in the opposite protein. This reinforces the *lock-and-key* structure within PPIs.

4.2 Contrastive Optimization

In addition to the the *lock-and-key* optimization objective, we aim to bring the residue features of the binding interface closer while pushing residue features that do not belong to the binding interface farther apart. To achieve this, we employ the contrastive loss [41] designed for point clouds. This loss minimizes the distance between the residue features of corresponding residues and maximizes the distance between non-corresponding residues as, $\mathcal{L}_{\text{contra}} = -\sum_{i \in S_R} \sum_{j \in S_L} \log \frac{\exp(\hat{\mathbf{f}}_R^i \cdot \hat{\mathbf{f}}_L^j / \tau_{\text{c}})}{\sum_{k \in S_L} \exp(\hat{\mathbf{f}}_R^i \cdot \hat{\mathbf{f}}_L^k / \tau_{\text{c}})}$. Here $S_R = (\mathbf{Y}_R^{\text{res}} = 1), S_L = (\mathbf{Y}_L^{\text{res}} = 1)$ are residues of binding interface for receptor and ligand, and τ_{c} is the temperature hyperparameter. This objective effectively minimizes the distance between residue features $\hat{\mathbf{f}}_R^i \in \hat{\mathbf{F}}_R, \hat{\mathbf{f}}_L^j \in \hat{\mathbf{F}}_L$ of corresponding binding interfaces, enhancing the proximity of relevant residue features.



(a) Interface dockQ similarity: **0.4845**

(b) Interface dockQ similarity: **0.4735**

(c) Interface dockQ similarity: **0.4572**

(d) Interface dockQ similarity: **0.5007**

(e) Interface dockQ similarity: **0.4550**

(f) Interface dockQ similarity: **0.5280**

Figure 2: Visualization of PPIretrieval results for proteins in the PDB test set, evaluated by *dockQ*. Proteins colored in blue are input query proteins; proteins colored in red are binding partners. Left column displays the ground-truth structures; right column shows the structure predictions.

4.3 Auxiliary Binding Interface Optimization

In addition to the *lock-and-key* and contrastive optimization objectives, we aim to make the predictions of binding interface $\hat{\mathbf{Y}}_R^{\text{res}}, \hat{\mathbf{Y}}_L^{\text{res}}$ close to the ground-truth values $\mathbf{Y}_R^{\text{res}} \in \{0,1\}^{N_R \times 1}, \mathbf{Y}_L^{\text{res}} \in \{0,1\}^{N_L \times 1}$, as $\mathcal{L}_{\text{bind}} = \texttt{BCE}(\hat{\mathbf{Y}}_R^{\text{res}}, \mathbf{Y}_R^{\text{res}}) + \texttt{BCE}(\hat{\mathbf{Y}}_L^{\text{res}}, \mathbf{Y}_L^{\text{res}})$. This objective ensures that the model makes accurate predictions for the binding interfaces by minimizing the difference between predicted and ground-truth values. The total loss is the sum of the three loss terms $\mathcal{L} = \mathcal{L}_{\text{match}} + \mathcal{L}_{\text{contra}} + \mathcal{L}_{\text{bind}}$. This optimization is designed to leverage the *lock-and-key* structure inherent in PPI complexes.

5 Retrieval Visualization

We visualize some retrieval results, showing the predictions of PPIretrieval when provided with an unseen query protein and observing its potential binding partner within our protein surface database. The database currently comprises a total of $151,207$ paired proteins with $302,414$ surface representations, trained and embedded using PPIs from PDB, DIPS, and PPBS training and validation sets (see Sec. 7). In Fig. 2, we observe that the predicted PPIs, with interface *dockQ* similarity, form a well-defined *lock-and-key* structure. This reliable structure formation bolsters confidence in the potential of PPIretrieval for exploring novel protein interactions. One can utilize our model and database to investigate and learn about unknown protein interactions. Additional visualizations are available in App. B.

6 Related Work

Protein Representation Learning The goal of protein representation learning is to derive meaningful and informative representations from protein sequences and structures. ESM [20] utilize protein sequences alongside evolutionary data to inform the learning process. SaProt [31] combine evolutionary insights with structure-aware protein tokens in its language modeling, achieving superior performance. GearNet [45] employs geometric contrastive learning to capture protein structural information. DSR [32] introduce a implicit neural network to represent protein surface representation. ProteinINR [18] leverage the multiview sequence-structure protein pretraining with additional implicit protein surface representations.

Protein-Protein Interactions Exploration Methods such as PPIformer [3] and MPAE-PPI [40] apply learned protein representations to explore protein-protein interactions. PPIformer [3] develops coarse-grained representations of protein complexes, defines structural masking of protein–protein interfaces to pretrain unlabeled PPIs. MPAE-PPI [40] encodes microenvironments into chemically meaningful discrete codes via a sufficiently large microenvironment vocabulary, and propose to capture the dependencies between different microenvironments.

7 Experiment

We use the PDB dataset from [9, 35], the DIPS dataset from [26], and the PPBS dataset from [36]. The PDB dataset comprises 4754 and 933 protein complexes for training and testing, , with 10% of the training set used for validation following [35]. The DIPS dataset includes 33159 and 8290 protein complexes for training and validation, with the first 4290 complexes of the validation set used for testing. The PPBS dataset comprises 101755, 10221, and 10911 protein complexes for training, validation, and testing (following homology split in [36]).

For surface sampling of each protein, we use dMaSIF [35] with sampling resolution 1.0Å, sampling distance 2.25Å, sampling number 20, and sampling variance 0.3Å. Additionally, we use 32 hidden dims and 0.3 dropout for all projections, use 2 propagation layers and 2 cross-attention layers. We choose a learning rate of $1e-4$ and use the AdamW optimizer with a weight decay of $5e-10$. We select the models with the lowest validation loss $\mathcal{L}$.

7.1 Empirical Evaluation

We empirically assess the quality of PPIs carried out by PPIretrieval during the inference stage. For PPIs in the test set, we measure the *dockQ* score [2], *TM* score [44], and root-mean-square-distance (*rmsd*) to compare PPIretrieval reference binding site, masif-predicted binding site [9], dmasif-predicted binding site [35], and PPIretrieval predicted binding site with the ground-truth binding sites. (1) ***dockQ* score** measures the quality between a ground-truth binding site and a predicted binding site, which combines the fraction of native contacts, the interface root mean square distance, and the ligand root mean square distance; a higher *dockQ* score indicates a better quality of the predicted binding site. (2) ***TM* score** measures the similarity between a ground-truth binding site and a predicted binding site, which considers both the distance and the alignment of the residues; a higher *TM* score indicates higher similarity between the two binding site. (3) ***rmsd*** measures the distance between a ground-truth binding site and a predicted binding site after superimposition; a lower *rmsd* indicates higher superimposition similarity between the two binding site. A comprehensive visualization and explanation of our metrics to assess quality of PPI binding sites are demonstrated in Fig. 3.

Dataset	Metrics		PDB	DIPS	PPBS
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4073	0.4177	0.5535
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.4220}$	$\underline{0.4304}$	$\underline{0.5946}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.1334	0.1021	0.1228
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.1155	0.0837	0.1036
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.2134	0.6617	0.4622
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.2366}$	$\underline{0.6649}$	$\underline{0.4735}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.0773	0.0981	0.0911
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.0665	0.0831	0.0871
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	11.40	11.33	$\underline{8.20}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{10.44}$	$\underline{6.02}$	9.77
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	15.73	19.66	17.32
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	17.87	23.55	19.65

Dataset	Metrics		PDB	DIPS	PPBS
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4250	0.4367	0.6014
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.4678}$	$\underline{0.4410}$	$\underline{0.6045}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.1345	0.1026	0.1249
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.1235	0.0998	0.1261
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.3222	0.6914	0.6014
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.3300}$	$\underline{0.6944}$	$\underline{0.6045}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.0833	0.1002	0.1004
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.0823	0.0911	0.1144
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	$\underline{9.30}$	9.65	9.96
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	10.70	$\underline{5.67}$	$\underline{6.52}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	15.56	19.05	16.82
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	16.81	21.22	16.08

Table 1: *dockQ*, *TM*, and *rmsd* for evaluation of **Top1 hit** binding sites predicted by PPIretrieval in comparison with other binding sites over three runs. ***Left:*** The database for each test set comprises surface features from the training and validation sets of each respective dataset. ***Right:*** The database comprises all surface features from the training and validation sets of PDB, DIPS, and PPBS datasets.

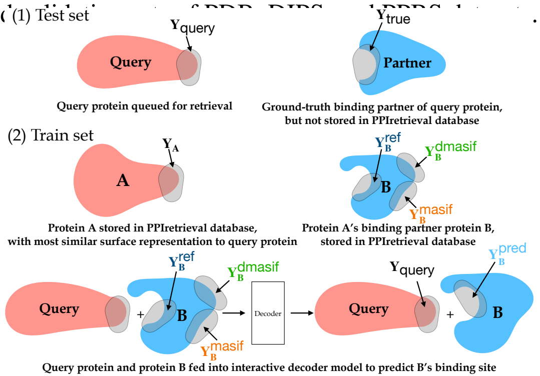

(1) Test set
$\mathbf{Y}_{\text{query}}$
Query
$\mathbf{Y}_{\text{true}}$
Partner
Query protein queued for retrieval
Ground-truth binding partner of query protein, but not stored in PPIretrieval database
(2) Train set
$\mathbf{Y}_{\mathbf{A}}$
A
$\mathbf{Y}_{\mathbf{B}}^{\text{ref}}$
$\mathbf{Y}_{\mathbf{B}}^{\text{dmasif}}$
B
$\mathbf{Y}_{\mathbf{B}}^{\text{masif}}$
Protein A stored in PPIretrieval database, with most similar surface representation to query protein
Protein A's binding partner protein B, stored in PPIretrieval database
Query
+
$\mathbf{Y}_{\mathbf{B}}^{\text{ref}}$
$\mathbf{Y}_{\mathbf{B}}^{\text{dmasif}}$
B
$\mathbf{Y}_{\mathbf{B}}^{\text{masif}}$
Decoder
$\mathbf{Y}_{\text{query}}$
Query
+
$\mathbf{Y}_{\mathbf{B}}^{\text{pred}}$
B
Query protein and protein B fed into interactive decoder model to predict B's binding site

Figure 3: Evaluation of PPI binding site during inference. For a PPI in the test set, a query protein with a known binding site $\mathbf{Y}_{\text{query}}$ seeks a binding partner with an actual binding site $\mathbf{Y}_{\text{true}}$. However, we assume that the binding partner is unknown to us. So, PPIretrieval aims to retrieve a potential binding partner from the surface databse. PPIretrieval identifies protein A in the surface database, which has the most similar surface representation to the query protein. Protein A has a known binding partner B with a reference binding site $\mathbf{Y}_B^{\text{ref}}$ (stored in database), a binding site $\mathbf{Y}_B^{\text{masif}}$ predicted by masif, and a binding site $\mathbf{Y}_B^{\text{dmasif}}$ predicted by dmasif. PPIretrieval takes query protein and B as input and predicts a new binding site $\mathbf{Y}_B^{\text{pred}}$. We compute *dockQ*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}})$, *TM*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}})$, *rmsd*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}})$, *dockQ*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}})$, *TM*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}})$, *rmsd*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}})$, *dockQ*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}})$, *TM*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}})$, *rmsd*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}})$, *dockQ*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}})$, *TM*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}})$, *rmsd*$(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}})$ to evaluate and compare the quality of PPI and binding interfaces. $\mathbf{Y}_{\text{true}}$ denotes the known binding site of the ground-truth binding partner; $\mathbf{Y}_B^{\text{ref}}$ denotes the known binding site (stored in database) of the retrieved binding partner; $\mathbf{Y}_B^{\text{pred}}$ denotes the predicted binding site of the retrieved binding partner.

Inference Result In Tab. 1-left, we assess the quality of PPIs and binding interface identified by PPIretrieval with smaller databases. The database for each test set comprises surface representations from the training set of the respective dataset. For example, when evaluating the PDB test set, we only search for surface representations in the PDB training set. Also, the models are trained on each respective dataset. In Tab. 1-right, we evaluate the quality of PPIs and binding interface identified by PPIretrieval with larger database. The database includes surface representations from the training and validation sets of the PDB, DIPS, and PPBS datasets, in total of $155,384$ paired proteins with their surface features for retrieval. And the models are trained on all training PPIs.

By comparing the tabular results, it is evident that the qualities of the predicted PPIs carried out by PPIretrieval improve with a larger database, as reflected in higher *dockQ* and *TM* scores, as well as lower *rmsd*. This improvement suggests that using PPIretrieval could be highly beneficial in facilitating the discovery of novel PPIs. More experimental results can be found in App. F

Cross-Dataset Validation for Generalization In addition, we show the cross-dataset performance in Tab. 2. We take the model trained on PDB training set only to encode the PPIs in DIPS and PPBS training and validation sets, respectively. Then we evaluate the cross-dataset performance on DIPS and PPBS test sets at two different hit rates. **Top10 hit** means that PPIretrieval retrieves the 10 most similar surface representations to the query protein in the test set for inference. Then, PPIretrieval decodes between the query protein and the 10 potential binders associated with these similar proteins. The best binding partner for the query protein is then selected based on the highest *dockQ* score.

The size and diversity of DIPS and PPBS exceed that of PDB, providing a robust test for generalization. The results, presented in Tab. 1, show improvements in dockQ and TM scores over the baseline results in Tab. 7.1, which are derived from models trained and tested within the same dataset. This enhancement in performance when applied to novel PPIs—unseen during training—affirms that our model has effectively learned to generalize the lock-and-key structures to new receptor-ligand pairs.

Dataset	Metrics		DIPS-Top1	DIPS-Top10	PPBS-Top1	PPBS-Top10
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4030	0.4156	0.5231	0.5611
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.4207}$	$\underline{0.4435}$	$\underline{0.5579}$	$\underline{0.5857}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.0515	0.0523	0.0621	0.0633
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.0434	0.0515	0.0601	0.0633
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.5330	0.6714	0.4202	0.3725
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.5419}$	$\underline{0.6792}$	$\underline{0.4421}$	$\underline{0.3889}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.0499	0.0511	0.0611	0.0620
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.0433	0.0491	0.0519	0.0602
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	11.12	$\underline{7.35}$	$\underline{8.92}$	$\underline{8.91}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{5.84}$	10.50	11.76	10.49
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	20.73	19.21	19.81	19.55
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	23.89	22.05	20.66.	20.04

Table 2: *dockQ*, *TM*, and *rmsd* for evaluation of **Top1, Top10 hit** binding sites predicted by PPIretrieval in comparison with other binding sites on cross-datasets over three runs. The database for each test set comprises surface representations from the training and validation sets of each respective dataset.

Computational Resources Our models are trained on a single Nvidia 48G A40 GPU. Regarding training time, PPIretrieval takes approximately 0.35s to train a protein complex. In terms of inference time, PPIretrieval requires about 0.11s for a protein complex.

7.2 Ablation Study

Model	PDB (dockQ ↑)	PDB (TM ↑)	PDB (rmsd ↓)
reference	0.4073	0.2134	11.40
surface-level heat-diffusion	$\underline{0.4220}$	$\underline{0.2366}$	$\underline{10.44}$
surface-level MPNN	0.4085	0.2254	11.21
surface-level GIN	0.4093	0.2265	11.26
residue-level heat-diffusion	0.3941	0.1901	12.13
residue-level MPNN	0.3435	0.1911	12.77
residue-level GIN	0.3566	0.2026	12.03

Model	PDB (dockQ ↑)	PDB (TM ↑)	PDB (rmsd ↓)	Time (↑)
reference	0.4073	0.2134	11.40	-
explicit cloud + geometric + chemical	$\underline{0.4220}$	$\underline{0.2366}$	$\underline{10.44}$	5 it/s
explicit cloud + chemical	0.3740	0.2001	12.38	6 it/s
explicit cloud + geometric	0.3375	0.1521	14.67	6 it/s
implicit representation + geometric + chemical	0.4113	0.2358	10.87	10 it/s
implicit representation + chemical	0.3439	0.1445	13.75	$\underline{\text{11 it/s}}$
implicit representation + geometric	0.3035	0.1301	14.89	$\underline{\text{11 it/s}}$

Table 3: Comparison of retrieval results with different baseline modules. ***Left:*** Surface-level heat-diffusion, MPNN [11], GIN [42] and residue-level heat-diffusion, MPNN [11], GIN [42] as encoders. ***Right:*** Explicit point cloud with geometric and chemical descriptors, and implicti neural representation [18, 32] with geometric and chemical descriptors.

7.2.1 Baseline Comparisons

In Tab. 3-left, we evaluate retrieval models equipped with various modules, including surface-level heat-diffusion, MPNN [11], GIN [42], and residue-level heat-diffusion, MPNN, GIN as encoders. Notably, encoders based on residue-level heat-diffusion, MPNN, and GIN yield neutral outcomes. In contrast, learning the lock-and-key structures and shape complementary between receptors and ligands is more effectively achieved at the surface level. Furthermore, our proposed surface encoder (at surface-level) outperforms MPNN and GIN in capturing the lock-and-key structures within PPI complexes. This capability to more precisely represent geometric and shape correspondences leads to enhanced generalization performance.

7.2.2 Implicit Surface Representation

In Tab. 3-right, we evaluate retrieval models using **implicit neural representations** [32, 18]. This involves training a signed distance function to represent protein surfaces, thereby eliminating the need for explicit representations. This aims to address memory-intensive issue and slow-inference associated with storing numerous protein complexes. We observe a trade-off between the dockQ similarity and inference speed when comparing explicit surfaces with implicit neural representations. It is beneficial to use implicit neural representations, for which we observe a significant increase in the inference time. Moreover, adopting implicit neural representations is more memory-efficient.

7.2.3 Training Objective & Model Design

Model	PDB (dockQ ↑)	PDB (TM ↑)	PDB (rmsd ↓)
reference	0.4073	0.2134	11.40
lock-and-key + contrastive + interface	$\underline{0.4220}$	$\underline{0.2366}$	$\underline{10.44}$
lock-and-key + contrastive	0.3991	0.2235	11.31
lock-and-key + interface	0.4173	0.2231	11.37
contrastive + interface	0.3591	0.1580	13.40
lock-and-key	0.3672	0.1938	12.55
contrastive	0.2044	0.1099	15.36
interface	0.3345	0.1567	15.21

Model	PDB (dockQ ↑)	PDB (TM ↑)	PDB (rmsd ↓)
reference	0.4073	0.2134	11.40
geometric + chemical + heat message-passing	$\underline{0.4220}$	$\underline{0.2366}$	$\underline{10.44}$
geometric + chemical + message-passing	0.4085	0.2254	11.21
geometric + message-passing	0.2938	0.1117	15.05
chemical + message-passing	0.3370	0.1839	13.30
geometric + heat message-passing	0.3375	0.1521	14.67
chemical + heat message-passing	0.3740	0.2001	12.38

Table 4: Comparison of retrieval results with different components. ***Left:*** Training objectives. ***Right:*** (Heat) message-passing with geometric and chemical descriptors.

In Tab. 4-left, we evaluate retrieval models with different training objectives. We observe that the lock-and-key and interface objectives contribute most to the model to learn the shape complementary between receptors and ligand, and the contrastive objective is a bonus loss for better generalization added to the lock-and-key and interface objectives.

In Tab. 4-right, we evaluate retrieval models with different model designs. We observe that the chemical descriptor and heat-MP contribute most to the model to learn the shape complementary within PPIs, the heat-MP can infer the geometric information existed in PPI complexes. However, it is also important to directly use geometric descriptor to provide the signals of geometric information to the model for learning the shape complementary between receptors and ligands.

7.3 Additional Experiments

PDB Dataset	Metrics		Top1	Top10	Top20	Top50	Top100
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4126	0.4331	0.4480	0.4491	0.4490
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.4235	0.4402	0.4531	0.4649	0.4708
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.1433	0.1436	0.1455	0.1458	0.1478
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.1225	0.1266	0.1301	0.1398	0.1405
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.3944	0.3877	0.3833	0.3554	0.3422
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.4041	0.3969	0.3863	0.3625	0.3528
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.0787	0.0766	0.0750	0.0721	0.0709
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.0536	0.0548	0.0588	0.0582	0.0601
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	10.41	9.73	9.70	9.49	9.32
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	10.04	8.97	8.66	8.20	7.35
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	15.73	15.71	15.54	15.26	15.19
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	17.75	17.22	17.02	16.40	16.11

Model	S cerevisiae (acc ↑)	S cerevisiae (roc ↑)	Human (acc ↑)	Human (roc ↑)
GCN LSTM-LM	91.42	95.26	97.93	98.37
GAT LSTM-LM	92.15	95.85	98.13	98.28
PPIretrieval LSTM-LM	94.30	97.22	98.99	99.03
GCN BERT-LM	86.68	92.01	96.32	97.59
GAT BERT-LM	86.74	92.23	96.59	97.35
PPIretrieval BERT-LM	86.89	92.55	97.31	98.44
GCN Onehot	71.09	79.23	81.30	84.25
GAT Onehot	69.23	75.27	79.84	82.24
PPIretrieval Onehot	74.56	82.21	82.57	86.00

Table 5: ***Left:*** *dockQ*, *TM*, and *rmsd* for evaluation of **Top1, Top10, Top20, Top50, Top100 hit** binding sites predicted by PPIretrieval in comparison with other binding sites in the PDB test set over three runs. ***Right:*** PPIretrieval for PPI classification in comparison with [17].

7.3.1 Hit Rates

In Tab. 5-left, we present experimental results for PPIretrieval at five different hit rates, increasing from **Top1** to **Top100**. The models are trained on all PDB, DIPS, and PPBS training set, and the database comprises surface representations from training and validation sets of them. We observe improved predicted interface quality in terms of *dockQ, TM* scores, and *rmsd*, with larger surface database. The robust experimental results suggest that PPIretrieval has the potential to facilitate and expedite the discovery of novel PPIs, identifying candidates with higher *dockQ* scores.

7.3.2 PPI classification

To underscore the distinctive benefits of our approach, we conduct experiments with existing PPI classification methods [17], emphasizing the use of surface-level features. This is done by adding a downstreaming head, thereby refining the model for PPI classification. In Tab. 5-right, we report the average results of 5-fold cross-validation following the approach in [17]. We observe improved accuracy and roc on both classification datasets with our method. By leveraging heat-MP for capturing surface features, our model can predict interactions based on structural complementary, which is a step beyond mere classification.

7.4 Case Study

In real-world scenarios, a protein can have a binding site that interacts with multiple partners. While some of these binding partners may be unknown, others might already be known and stored in our database. PPIretrieval can be effectively used to identify these binding partners by finding similar surface representations within the database. A case study demonstration is visualized in Fig. 4.

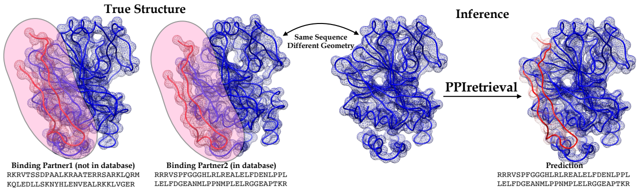

True Structure
Inference
Same Sequence
Different Geometry
PPIretrieval
Binding Partner1 (not in database)
RKRVTSSDPAALKRAATERRSARKLQRM
KQLEDLLSKNYHLENVEALRKKLVGER
Binding Partner2 (in database)
RRRVSPFGGGHLRLREALELFDENLPPL
LELFDGEANMLPPNMPLELRGGEAPTKR
Prediction
RRRVSPFGGGHLRLREALELFDENLPPL
LELFDGEANMLPPNMPLELRGGEAPTKR

Figure 4: Case study using PPIretrieval. The query protein, highlighted in blue, successfully identifies a binding partner within our surface database using PPIsearch.

In our case study, the query protein has two binding partners: one is already stored in our surface database (*pdb id: 5J28*), while the other is not (*pdb id: 1DGC*). It is important to note that, although the query protein in the two ground-truth structures shares the same sequence representation, there are slight differences in their geometric configuration. Given the query protein, PPIretrieval identifies the protein in the database that most closely matches in both sequential and geometric representation. Thus, it successfully predicts the corresponding binding partner for this query protein.

8 Limitation and Future Work

Indeed, the storage of thousands of surface representations of protein complexes can be memory-intensive, potentially requiring several gigabytes of space. To mitigate this issue, we will use implicit neural networks for protein surface representations [32, 3]. This involves training a signed distance function to represent the protein surfaces, which eliminates the need to store explicit surface representations. With implicit representation, a potential work is to train a larger model with increased parameters to better approximate the protein surface manifold. Also, we aim to continuously integrate more high-quality PPI data into the collection of our database. As demonstrated in Tab. 2, PPIretrieval exhibits the ability to generalize to unseen proteins. Therefore, future work involves training our model on new PPI data, enabling direct encoding and storage of these data in our existing database.

References

[1] A. Athanasios, V. Charalampos, T. Vasileios, and G. Md Ashraf. Protein-protein interaction (ppi) network: recent advances in drug discovery. *Current drug metabolism*, 18(1):5–10, 2017.

[2] S. Basu and B. Wallner. Dockq: a quality measure for protein-protein docking models. *PloS one*, 11(8):e0161879, 2016.

[3] A. Bushuiev, R. Bushuiev, A. Filkin, P. Kouba, M. Gabrielova, M. Gabriel, J. Sedlar, T. Pluskal, J. Damborsky, S. Mazurenko, et al. Learning to design protein-protein interactions with enhanced generalization. *arXiv preprint arXiv:2310.18515*, 2023.

[4] H. Cai, C. Shen, T. Jian, X. Zhang, T. Chen, X. Han, Z. Yang, W. Dang, C.-Y. Hsieh, Y. Kang, et al. Carsidock: a deep learning paradigm for accurate protein–ligand docking and screening based on large-scale pre-training. *Chemical Science*, 15(4):1449–1471, 2024.

[5] Y. Cao, D. Li, H. Sun, A. H. Assadi, and S. Zhang. Efficient curvature estimation for oriented point clouds. *stat*, 1050:26, 2019.

[6] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[7] J. De Las Rivas and C. Fontanillo. Protein–protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS computational biology*, 6(6):e1000807, 2010.

[8] D. C. Fry. Protein–protein interactions as targets for small molecule drug discovery. *Peptide Science: Original Research on Biomolecules*, 84(6):535–552, 2006.

[9] 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):184–192, 2020.

[10] Z. Gao, C. Jiang, J. Zhang, X. Jiang, L. Li, P. Zhao, H. Yang, Y. Huang, and J. Li. Hierarchical graph learning for protein–protein interaction. *Nature Communications*, 14(1):1093, 2023.

[11] 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*, pages 1263–1272. PMLR, 2017.

[12] V. Gligorijević, P. D. Renfrew, T. Kosciolek, J. K. Leman, D. Berenberg, T. Vatanen, C. Chandler, B. C. Taylor, I. M. Fisk, H. Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

[13] C. Hua, S. Luan, M. Xu, R. Ying, J. Fu, S. Ermon, and D. Precup. Mudiff: Unified diffusion for complete molecule generation. *arXiv preprint arXiv:2304.14621*, 2023.

[14] C. Hua, G. Rabusseau, and J. Tang. High-order pooling for graph neural networks with tensor decomposition. *Advances in Neural Information Processing Systems*, 35:6021–6033, 2022.

[15] C. Isert, K. Atz, and G. Schneider. Structure-based drug design with geometric deep learning. *Current Opinion in Structural Biology*, 79:102548, 2023.

[16] V. Jain and H. Zhang. Robust 3d shape correspondence in the spectral domain. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 19–19. IEEE, 2006.

[17] K. Jha, S. Saha, and H. Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):8360, 2022.

[18] Y. Lee, H. Yu, J. Lee, and J. Kim. Pre-training sequence, structure, and surface features for comprehensive protein representation learning. In *The Twelfth International Conference on Learning Representations*, 2023.

[19] B. Lévy. Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13. IEEE, 2006.

[20] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[21] J. Lu, Y. Sun, and Q. Huang. Jigsaw: Learning to assemble multiple fractured objects. *arXiv preprint arXiv:2305.17975*, 2023.

[22] S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

[23] S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375, 2022.

[24] S. Luan, M. Zhao, C. Hua, X.-W. Chang, and D. Precup. Complete the missing half: Augmenting aggregation filtering with diversification for graph convolutional networks. *arXiv preprint arXiv:2008.08844*, 2020.

[25] M. McPartlon and J. Xu. Deep learning for flexible and site-specific protein docking and design. *BioRxiv*, pages 2023–04, 2023.

[26] A. Morehead, C. Chen, A. Sedova, and J. Cheng. Dips-plus: The enhanced database of interacting protein structures for interface prediction. *Scientific Data*, 10(1):509, 2023.

[27] J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert. A lock-and-key model for protein–protein interactions. *Bioinformatics*, 22(16):2012–2019, 2006.

[28] M. Réau, N. Renaud, L. C. Xue, and A. M. Bonvin. Deeprank-gnn: a graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics*, 39(1):btac759, 2023.

[29] V. G. Satorras, E. Hoogeboom, and M. Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[30] D. E. Scott, A. R. Bayly, C. Abell, and J. Skidmore. Small molecules, big targets: drug discovery faces the protein–protein interaction challenge. *Nature Reviews Drug Discovery*, 15(8):533–550, 2016.

[31] J. Su, C. Han, Y. Zhou, J. Shan, X. Zhou, and F. Yuan. Saprot: protein language modeling with structure-aware vocabulary. *bioRxiv*, pages 2023–10, 2023.

[32] D. Sun, H. Huang, Y. Li, X. Gong, and Q. Ye. Dsr: Dynamical surface representation as implicit neural networks for protein. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, pages 1383–1392. Wiley Online Library, 2009.

[34] T. Sun, B. Zhou, L. Lai, and J. Pei. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC bioinformatics*, 18:1–8, 2017.

[35] F. Sverrisson, J. Feydy, B. E. Correia, and M. M. Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15272–15281, 2021.

[36] J. Tubiana, D. Schneidman-Duhovny, and H. J. Wolfson. Scannet: an interpretable geometric deep learning model for structure-based protein binding site prediction. *Nature Methods*, 19(6):730–739, 2022.

[37] M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, J. Lee, C. L. Gilchrist, J. Söding, and M. Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, pages 1–4, 2023.

[38] Y. Wang, Y. Shen, S. Chen, L. Wang, F. Ye, and H. Zhou. Learning harmonic molecular representations on riemannian manifold. *arXiv preprint arXiv:2303.15520*, 2023.

[39] D. Whitford. *Proteins: structure and function*. John Wiley & Sons, 2013.

[40] L. Wu, Y. Tian, Y. Huang, S. Li, H. Lin, N. V. Chawla, and S. Z. Li. Mape-ppi: Towards effective and efficient protein-protein interaction prediction via microenvironment-aware protein embedding. *arXiv preprint arXiv:2402.14391*, 2024.

[41] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020.

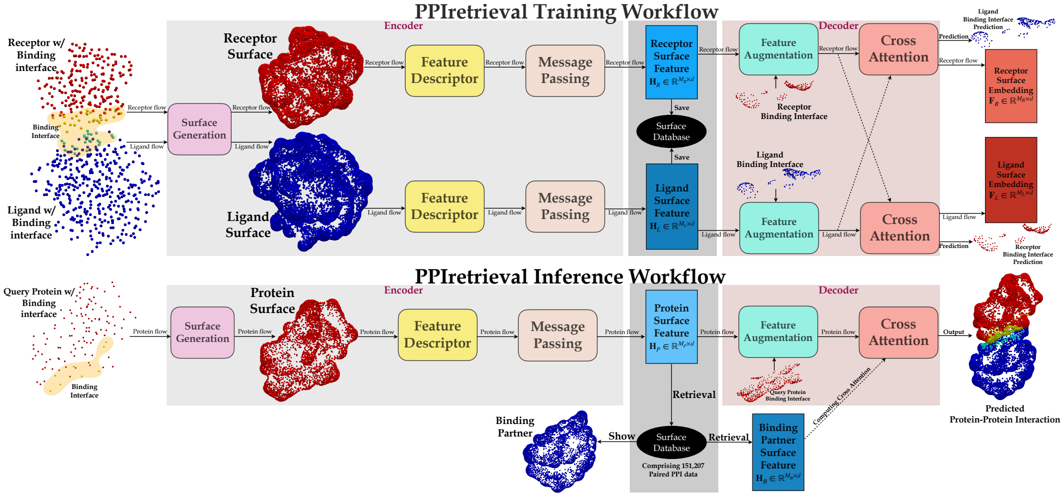
[42] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[43] X. Zhang, O. Zhang, C. Shen, W. Qu, S. Chen, H. Cao, Y. Kang, Z. Wang, E. Wang, J. Zhang, et al. Efficient and accurate large library ligand docking with karmadock. *Nature Computational Science*, 3(9):789–804, 2023.

[44] Y. Zhang and J. Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

[45] Z. Zhang, M. Xu, A. Jamasb, V. Chenthamarakshan, A. Lozano, P. Das, and J. Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

A PPIretrieval Model Design



PPIretrieval Training Workflow
Encoder
Receptor w/ Binding interface
Binding Interface
Ligand w/ Binding interface
Receptor flow
Ligand flow
Surface Generation
Receptor Surface
Ligand Surface
Feature Descriptor
Message Passing
Receptor Surface Feature $\mathbf{H}_R \in \mathbb{R}^{M_R \times d}$
Ligand Surface Feature $\mathbf{H}_L \in \mathbb{R}^{M_L \times d}$
Save
Surface Database
Decoder
Feature Augmentation
Receptor Binding Interface
Ligand Binding Interface
Cross Attention
Prediction
Ligand Binding Interface Prediction
Receptor Surface Embedding $\mathbf{F}_R \in \mathbb{R}^{M_R \times d}$
Ligand Surface Embedding $\mathbf{F}_L \in \mathbb{R}^{M_L \times d}$
Receptor Binding Interface Prediction
PPIretrieval Inference Workflow
Query Protein w/ Binding interface
Binding Interface
Protein flow
Protein Surface
Protein Surface Feature $\mathbf{H}_P \in \mathbb{R}^{M_P \times d}$
Retrieval
Query Protein Binding Interface
Output
Predicted Protein-Protein Interaction
Binding Partner
Show
Comprising 151,207 Paired PPI data
Binding Partner Surface Feature $\mathbf{H}_B \in \mathbb{R}^{M_B \times d}$
Computing Cross Attention

Figure 5: An overview of PPIretrieval pipeline, demonstrating the training and inference workflows. **During training**, PPIretrieval processes a PPI complex. The encoder network (shown in grey) encodes the two proteins, generating surface features, $\mathbf{H}_R, \mathbf{H}_L$. These features, along with information about their binding partnership, are stored in our database. The decoder network (shown in brown) then takes these surface features, along with the receptor's binding interface as input, predicting the ligand's binding interface and generating its embedding $\mathbf{F}_L$, and vice versa. **During inference**, PPIretrieval takes a protein P with its corresponding binding interface to the encoder network. It encodes P into a surface feature $\mathbf{H}_P$. Then, PPIretrieval identifies a surface feature $\mathbf{H}_B$ for binding partner B in our database. The decoder network takes the surface features $\mathbf{H}_P, \mathbf{H}_B$, along with P's binding interface, predicting B's binding interface. A PPI complex is predicted between the input protein P with the given binding interface and the predicted binding partner B with the predicted binding interface, as demonstrated above. Details of PPIretrieval can be found in Sec. 3.

The overview of PPIretrieval is demonstrated in Fig. 1. PPIretrieval follows a specific design process. Starting with a query protein P comprising N_P residues in Euclidean space $\upsilon_P \in \mathbb{R}^{N_P \times 3}$, along with its corresponding binding interface $\mathbf{Y}_P^{\text{res}}$, we first encode the protein into a surface representation $\mathbf{H}_P \in \mathbb{R}^{M_P \times d}$. The next step involves retrieving the most similar surface feature $\mathbf{H}_A \in \mathbb{R}^{M_A \times d}$ in the database. Once identified, we locate a binding partner protein B that binds to A, with a surface feature $\mathbf{H}_B \in \mathbb{R}^{M_B \times d}$. Then, the model decodes the pair of protein surface features, $\mathbf{H}_P, \mathbf{H}_B$, utilizing the known binding interface $\mathbf{Y}_P^{\text{res}}$. The final prediction involves estimating the binding interface $\hat{\mathbf{Y}}_B^{\text{res}}$ for protein B. In summary, the model outputs the protein B that is most likely to bind to the input protein P, accompanied by the predicted binding interface $\hat{\mathbf{Y}}_B^{\text{res}}$.

It is important to highlight that our entire pipeline operates on a deep-learning framework. There is no need for precomputation of protein or surface patches, making PPIretrieval an efficient tool. This characteristic enables a fast encoding for the input protein and facilitates effective retrieval for the output protein within the model.

A.1 Surface Representation Encoder

The surface encoder network aims to encode an input protein P into a surface representation $\mathbf{H}_P \in \mathbb{R}^{M_P \times d}$, where M_P denotes the number of surface points representing P. This representation captures the propagated chemical and geometric information. It can be stored in our database for subsequent retrieval and comparison purposes.

Protein Surface Preparation The input consists of one-hot encoded residue types for protein $P \in [0,1]^{N_P \times 20}$ in Euclidean space $\upsilon_P \in \mathbb{R}^{N_P \times 3}$, along with the corresponding binding interface $\mathbf{Y}_P^{\text{res}} \in \{0,1\}^{N_P \times 1}$. We apply dMaSIF [35] to generate a set of oriented surface points $\{x_1, ..., x_{M_P}\} \in \mathbb{R}^{M_P \times 3}$ with unit normals $\{n_1, ..., n_{M_P}\} \in \mathbb{R}^{M_P \times 3}$ to approximate a smooth manifold representing the surface of protein P.

Upon approximating the protein surface with surface points, we define the LB operator Δ_P on the surface for heat-MP. We compute the first k eigenfunctions of Δ_P stacked in matrix $\Phi_P \in \mathbb{R}^{M\times k}$ with their corresponding eigenvalues $\{\lambda_i\}_{i=1}^{k}$ ($k=100$), then calculate the Moore-Penrose pseudo-inverse of the eigenfunction matrix, $\Phi_P^{+} \in \mathbb{R}^{k\times M}$. The heat-MP allows effective information flow between surface points, crucial for operating heat-MP on the surface [38].

Geometric Descriptor Following the computation of surface points to represent the input protein, we describe the local geometric features of the surface. We approximate the per-point mean curvature, Gaussian curvature as detailed in [5], and compute the Heat Kernel Signatures as described in [33]. The geometric features processed by a MLP, $\mathbf{F}_{\text{Geom}} \leftarrow \text{MLP}([\mathbf{F}_{\text{Mean}}, \mathbf{F}_{\text{Gauss}}, \mathbf{F}_{\text{HKS}}]) \in \mathbb{R}^{M_P\times d_G}$, capture the local geometric environment for each surface point.

Chemical Descriptor We proceed to compute chemical features for surface points based on the one-hot encoded residue types and the binding interface. The residue-level chemical features are first encoded using a MLP with concatenated features, $\mathbf{F}_{\text{Res}} \leftarrow \text{MLP}([P, \mathbf{Y}_P^{\text{res}}]) \in \mathbb{R}^{N_P\times d_C}$, then transformed by an equivariant-GNN [29], $\mathbf{F}_{\text{Res}} \leftarrow \text{EGNN}(\mathbf{F}_{\text{Res}}, \upsilon_P) \in \mathbb{R}^{N_P\times d_C}$. The residue-level features $\mathbf{F}_{\text{Res}}$ preserve the local information pertaining to residues and the binding interface.

Then, we project these residue-level features onto surface-level features. For each surface point x_i, we identify its k nearest neighboring residues $\{r_1^i, ..., r_k^i\}$ with features $\{\mathbf{f}_{\text{Res}}^{i,1}, ..., \mathbf{f}_{\text{Res}}^{i,k}\}$. A vector of chemical features is computed by applying a MLP and summation over the nearest neighboring residues with a distance filter[1], $\mathbf{f}_{\text{Chem}}^{i} \leftarrow \text{MLP}(\sum_{j=1}^{k} f_{\cos}(\|x_i - r_j^i\|) \cdot \text{MLP}([\mathbf{f}_{\text{Res}}^{i,j}, 1/\|x_i - r_j^i\|])) \in \mathbb{R}^{d_C}$. By computing and stacking these per-point features, we obtain the chemical features for the protein surface $\mathbf{F}_{\text{Chem}} = \{\mathbf{f}_{\text{Chem}}^{1}, ..., \mathbf{f}_{\text{Chem}}^{M}\} \in \mathbb{R}^{M_P\times d_C}$.

Finally, we use a MLP to combine the per-point geometric and chemical features, $\mathbf{F}_{\text{Surf}} \leftarrow \text{MLP}([\mathbf{F}_{\text{Geom}}, \mathbf{F}_{\text{Chem}}]) \in \mathbb{R}^{M_P\times d}$. For each surface point, these features capture the local geometric and chemical environment, along with binding interface information. These can be effectively used in the heat-MP for facilitating information flow on the approximated protein surface.

Message-passing on Protein Surface We perform heat-MP on protein surfaces for message passing. This operation can be smoothly applied to the protein surface, treating it as a compact object [38]. For a protein with surface feature $\mathbf{F}_{\text{Surf}}$, we first project it onto the column space of eigenfunctions, $\mathbf{F}'_{\text{Surf}} \leftarrow \Phi_P^{+}\mathbf{F}_{\text{Surf}} \in \mathbb{R}^{k\times d}$, expressing features in the orthogonal basis with a reduced dimension. Then following Eq. 3, we can perform heat-MP on the surface as,

$$\mathbf{H}_P = \Phi \underbrace{\begin{pmatrix} e^{-\frac{(\lambda_1-\mu_1)^2}{\sigma_1^2}-\lambda_1 t_1} & \cdots & e^{-\frac{(\lambda_1-\mu_d)^2}{\sigma_d^2}-\lambda_1 t_d} \\ \vdots & \ddots & \vdots \\ e^{-\frac{(\lambda_k-\mu_1)^2}{\sigma_1^2}-\lambda_k t_1} & \cdots & e^{-\frac{(\lambda_k-\mu_d)^2}{\sigma_d^2}-\lambda_k t_d} \end{pmatrix}}_{=F\exp(-\lambda t)\in\mathbb{R}^{k\times d}} \odot \mathbf{F}'_{\text{Surf}} \in \mathbb{R}^{M\times d}. \tag{4}$$

Here, each feature channel of $\mathbf{F}_{\text{Surf}}$ has its unique set of $\{\mu_i, \sigma_i, t_i\}_{i=1}^{d}$. In summary, the encoder network encodes a protein (represented by residues) into a surface representation by applying heat-MP on the surface, capturing both local geometric and chemical environments.

Training Stage During training, the encoder network is fed with the paired receptor and ligand proteins R, L, along with their corresponding binding interfaces $\mathbf{Y}_R^{\text{res}}, \mathbf{Y}_L^{\text{res}}$. It then processes these inputs to generate distinct surface features for each protein,

$$\mathbf{H}_R = \text{Enc}(R, \mathbf{Y}_R^{\text{res}}) \in \mathbb{R}^{M_R\times d},\ \mathbf{H}_L = \text{Enc}(L, \mathbf{Y}_L^{\text{res}}) \in \mathbb{R}^{M_L\times d}. \tag{5}$$

Binding Interface of Surface We construct the surface-level binding interface $\mathbf{Y}_P^{\text{surf}} \in \{0,1\}^{M_P\times 1}$ for surface points based on the residue-level binding interface $\mathbf{Y}_P^{\text{res}} \in \{0,1\}^{N_P\times 1}$ for protein P. Here, $\mathbf{y}_{i,P}^{\text{surf}} = 1$ indicates that the surface point x_i belongs to the binding interface. For each residue in P, we define a region with a predefined radius of $r = 10\mathring{\text{A}}$. All surface points falling within this region are then labeled as part of the surface binding interface.

[1] We use a cosine cutoff function, $f_{\cos}(d) = \frac{1}{2}(\cos(\frac{\pi d}{d_{\text{cut}}}) + 1)$, to smooth out the distance transition to 0 as the distance d approaches a pre-defined cutoff distance $d_{\text{cut}} = 30\mathring{\text{A}}$.

A.2 Interactive Decoder

The decoder network operates by taking surface features as input, allowing interaction between two proteins, and ultimately predicting a binding interface.

We assume that input data comprises the receptor R and its corresponding binding interface $\mathbf{Y}_R^{\text{res}}$. The encoder network generates a surface feature $\mathbf{H}_R \in \mathbb{R}^{M_R \times d}$. Simultaneously, we compute the surface-level binding interface $\mathbf{Y}_R^{\text{surf}} \in \{0,1\}^{M_R \times 1}$. Then, the model identifies a binding partner L and obtains its own surface feature $\mathbf{H}_L \in \mathbb{R}^{M_L \times d}$. The objective is to predict the binding interface for ligand L, expressed as $p(\hat{\mathbf{Y}}_L^{\text{res}} | \mathbf{H}_R, \mathbf{H}_L, \mathbf{Y}_R^{\text{surf}}) \in [0,1]^{N_L \times 1}$.

Cross-Attention Before computing the cross-attention, we update the surface features for receptor and ligand using an equivariant-GNN and MLPs, $\mathbf{H}_R \leftarrow \text{EGNN}(\text{MLP}([\mathbf{H}_R, \mathbf{Y}_R^{\text{surf}}]), x_R) \in \mathbb{R}^{M_R \times d}, \mathbf{H}_L \leftarrow \text{EGNN}(\text{MLP}(\mathbf{H}_L), x_L) \in \mathbb{R}^{M_L \times d}$. Here, $\mathbf{H}_R$ is updated with information about the binding interface, providing the model with improved capabilities for locating the binding interface of the ligand L during cross-attention.

Given the updated receptor features $\mathbf{H}_R$ and ligand features $\mathbf{H}_L$, we compute the cross-attention between two protein surface features, enabling interaction and communication,

$$\mathbf{F}_R = \text{softmax}\left(\frac{(\mathbf{H}_R W_{\text{Q}})(\mathbf{H}_L W_{\text{K}})^T}{\sqrt{d}}\right)(\mathbf{H}_L W_{\text{V}}),$$

$$\mathbf{F}_L = \text{softmax}\left(\frac{(\mathbf{H}_L W_{\text{Q}})(\mathbf{H}_R W_{\text{K}})^T}{\sqrt{d}}\right)(\mathbf{H}_R W_{\text{V}}),$$

where $W_{\text{Q}}, W_{\text{K}}, W_{\text{V}} \in \mathbb{R}^{d \times d}$ are learned parameters for the query, key, value in the attention mechanism, respectively. This facilitates effective interaction and communication between the receptor and ligand surface features.

Binding Interface Once we obtain the propagated surface features for the ligand $\mathbf{F}_L \in \mathbb{R}^{M_L \times d}$, we employ a MLP with sigmoid function directly on these features, predicting for the binding interface, $\hat{\mathbf{Y}}_L^{\text{surf}} \leftarrow \sigma(\text{MLP}(\mathbf{F}_L)) \in [0,1]^{M_L \times 1}$. Thus, the prediction of surface-level binding interface for ligand L is solely conditioned by the surface features of both the receptor and the ligand, along with the binding interface information of the receptor.

Surface Point to Residue We compute the residue-level binding interface $\hat{\mathbf{Y}}_L^{\text{res}} \in \{0,1\}^{N_L \times 1}$ from $\hat{\mathbf{Y}}_L^{\text{surf}}$, and embedding $\hat{\mathbf{F}}_L \in \mathbb{R}^{N_L \times d}$ from $\mathbf{F}_L$. For each residue i in L, we define a region with a fixed radius of $r = 10\text{\AA}$ and collect a set of surface points within this region, each with a binding interface $\hat{\mathbf{y}}_j^{\text{surf}}$ and embedding $\mathbf{f}_j$. The residue i is considered part of the binding interface if the majority of surface points in the region are labeled as part of the binding interface, i.e., $\hat{\mathbf{y}}_i^{\text{res}} = 1$ if $\text{Mean}(\sum_j \hat{\mathbf{y}}_j^{\text{surf}}) > 0.5$; otherwise $\hat{\mathbf{y}}_i^{\text{res}} = 0$. And the residue-level embedding is obtained using the same logic, where $\hat{\mathbf{f}}_i = \text{Mean}(\sum_j \mathbf{f}_j)$.

Training Stage During training, each PPI sample is treated as two training instances. The model first takes the receptor R and its associated binding interface $\mathbf{Y}_R^{\text{surf}}$ as input, predicting the ligand's binding interface; then the model takes the ligand L and its corresponding binding interface $\mathbf{Y}_L^{\text{surf}}$ as input, predicting the receptor's binding interface,

$$\hat{\mathbf{Y}}_L^{\text{res}}, \hat{\mathbf{F}}_L = \text{Dec}(\mathbf{H}_R, \mathbf{H}_L, \mathbf{Y}_R^{\text{surf}}),\ \hat{\mathbf{Y}}_R^{\text{res}}, \hat{\mathbf{F}}_R = \text{Dec}(\mathbf{H}_R, \mathbf{H}_L, \mathbf{Y}_L^{\text{surf}}). \tag{6}$$

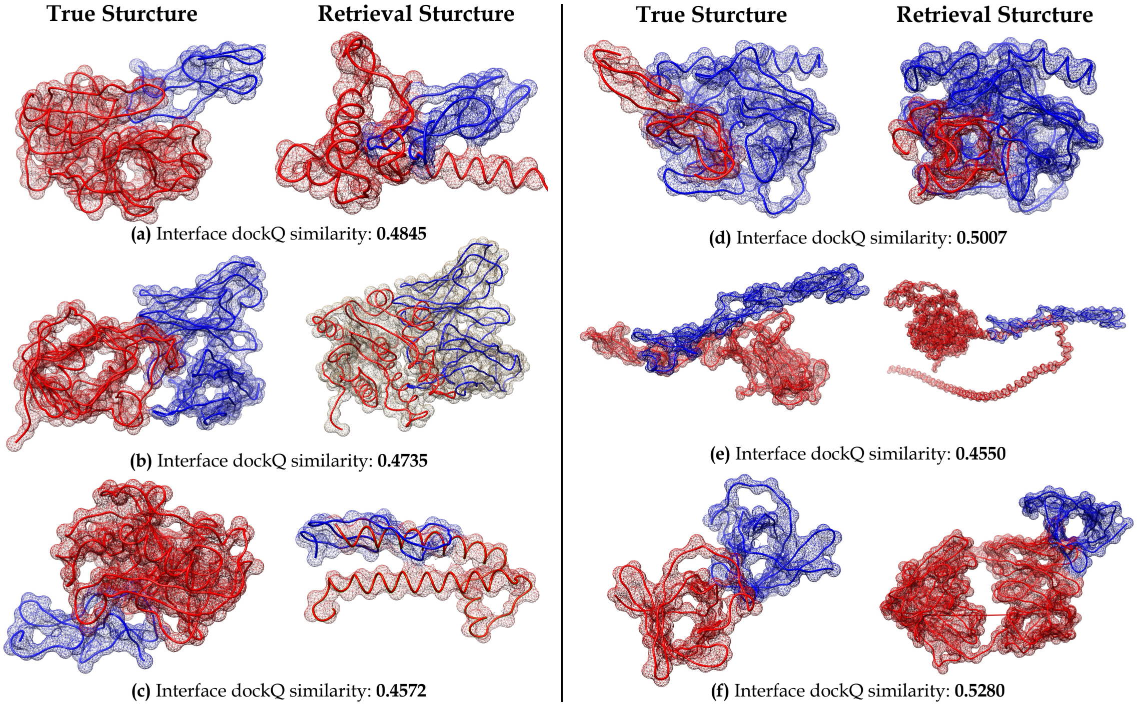
A.3 Overview: PPIretrieval Pipeline

Training Stage During training, PPIretrieval processes a PPI complex with their corresponding binding interface. The encoder network encodes the two proteins, resulting in two surface features $\mathbf{H}_R, \mathbf{H}_L$, respectively. These surface representations, along with information about their binding partnership, are stored in our database. Then, the decoder network takes the surface features, $\mathbf{H}_R, \mathbf{H}_L$, along with the receptor's binding interface $\mathbf{Y}_R^{\text{surf}}$ as input, and predicts the ligand's binding interface $\hat{\mathbf{Y}}_L^{\text{res}}$. This process is repeated to predict the receptor's binding interface $\hat{\mathbf{Y}}_R^{\text{res}}$ vice versa. PPIretrieval undergoes optimization to utilize the *lock-and-key* structures of the PPI complex, following a specific approach discussed in Sec. 4. This training setup ensures that PPIretrieval learns to predict binding interfaces for both proteins involved in a PPI complex.

Inference Stage During inference, PPIretrieval takes an unseen query protein P along with a specified binding interface to the encoder network. It encodes P into a surface feature $\mathbf{H}_P$. Then, PPIretrieval searches our database to retrieve the most similar surface feature $\mathbf{H}_A$ using a similarity function. Once the match is found, the surface feature $\mathbf{H}_B$ of the binding partner B that binds to A is identified. Then, the decoder network takes the surface features $\mathbf{H}_P, \mathbf{H}_B$, along with P's given binding interface $\mathbf{Y}_P^{\text{surf}}$, and predicts B's binding interface, $\hat{\mathbf{Y}}_B^{\text{res}}$. This process entails the model making predictions on how the surface of the binding partner interacts with the provided protein and its binding interface. Finally, PPIretrieval outputs protein B along with the predicted binding interface, indicating the most likely binding scenario with protein P and its given binding interface. A visual demonstration of this process is illustrated in Fig. 1.

B Protein-Protein Interaction Visualization

We visualize additional predicted PPIs carried out by PPIretrieval.



(a) Interface dockQ similarity: **0.4845**

(b) Interface dockQ similarity: **0.4735**

(c) Interface dockQ similarity: **0.4572**

(d) Interface dockQ similarity: **0.5007**

(e) Interface dockQ similarity: **0.4550**

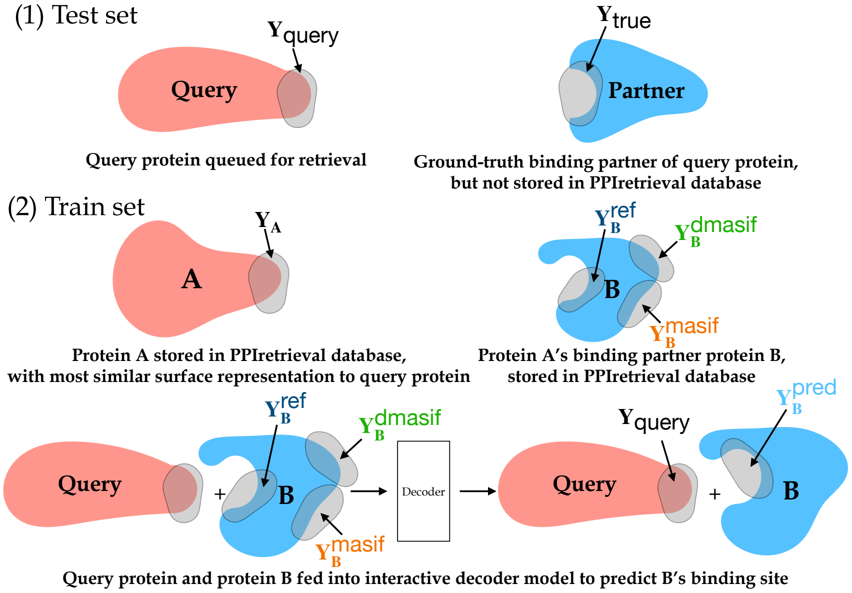
(f) Interface dockQ similarity: **0.5280**

Figure 6: Visualization of PPIretrieval results for proteins in the PDB test set, evaluated by *dockQ*. Proteins colored in blue are input query proteins; proteins colored in red are binding partners. Left column displays the ground-truth structures; right column shows the structures predicted by PPIretrieval.

In Fig. 6, we observe that the predicted PPIs, with interface *dockQ* similarity, form a well-defined *lock-and-key* structure. This reliable structure formation bolsters confidence in the potential of PPIretrieval for exploring novel protein interactions. One can utilize our model and database to investigate and learn about unknown protein interactions.

C Additional Experiments on Empirical Evaluation

In additional to $\textit{dockQ}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}), \textit{TM}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}), \textit{rmsd}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}})$, $\textit{dockQ}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}), \textit{TM}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}), \textit{rmsd}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}), \textit{dockQ}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}), \textit{TM}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}), \textit{rmsd}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}})$ used in Sec 7, we evaluate $\textit{dockQ}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}), \textit{TM}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}), \textit{rmsd}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}})$, $\textit{dockQ}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}), \textit{TM}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}), \textit{rmsd}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}), \textit{dockQ}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}), \textit{TM}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}), \textit{rmsd}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}})$as demonstrated in Fig. 7


(1) Test set
$\mathbf{Y}_{\text{query}}$
Query
$\mathbf{Y}_{\text{true}}$
Partner
(2) Train set
$\mathbf{Y}_{\mathbf{A}}$
A
$\mathbf{Y}_{\mathbf{B}}^{\text{ref}}$
$\mathbf{Y}_{\mathbf{B}}^{\text{dmasif}}$
B
$\mathbf{Y}_{\mathbf{B}}^{\text{masif}}$
$\mathbf{Y}_{\mathbf{B}}^{\text{ref}}$
$\mathbf{Y}_{\mathbf{B}}^{\text{dmasif}}$
Query
+
B
$\mathbf{Y}_{\mathbf{B}}^{\text{masif}}$
Decoder
$\mathbf{Y}_{\text{query}}$
$\mathbf{Y}_{\mathbf{B}}^{\text{pred}}$
Query
+
B

Query protein queued for retrieval

Ground-truth binding partner of query protein, but not stored in PPIretrieval database

Protein A stored in PPIretrieval database, with most similar surface representation to query protein

Protein A's binding partner protein B, stored in PPIretrieval database

Query protein and protein B fed into interactive decoder model to predict B's binding site

Figure 7: Evaluation of PPI quality during inference. For a PPI in the test set, a query protein with a known binding site $\mathbf{Y}_{\text{query}}$ seeks a binding partner with an actual binding site $\mathbf{Y}_{\text{true}}$. We compute $\textit{dockQ}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}), \textit{TM}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}), \textit{rmsd}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}})$ to measure quality of the ground-truth PPI. However, the binding partner is unknown to the PPIretrieval surface database. So, PPIretrieval aims to find a potential binding partner from the surface databse. PPIretrieval identifies protein A in the surface database, which has the most similar surface representation to the query protein. Protein A has a known binding partner B with a binding site $\mathbf{Y}_B^{\text{ref}}$. We compute $\textit{dockQ}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}), \textit{TM}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}), \textit{rmsd}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}})$ to assess quality of the PPI, and $\textit{TM}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}), \textit{rmsd}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}})$ to evaluate quality of the binding sites. PPIretrieval takes query protein and B as input and predicts a new binding site $\mathbf{Y}_B^{\text{pred}}$. We compute $\textit{dockQ}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}), \textit{TM}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}), \textit{rmsd}(\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}})$ to assess quality of the PPI, $\textit{dockQ}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}), \textit{TM}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}), \textit{rmsd}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}})$, $\textit{dockQ}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}), \textit{TM}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}), \textit{rmsd}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}})$, $\textit{dockQ}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}})$, $\textit{TM}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}})$, $\textit{rmsd}(\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}})$ to evaluate and compare quality of the binding sites.

In Tab. 6, we assess the quality of PPIs and binding sites identified by PPIretrieval with smaller databases. The database for each test set comprises surface representations from the training set of the respective dataset. For example, when evaluating the PDB test set, we only search for surface representations in the PDB training set. Also, the models are trained on each respective dataset.

In Tab. 7, we evaluate the quality of PPIs and binding sites identified by PPIretrieval with larger database. The database includes surface representations from the training and validation sets of the PDB, DIPS, and PPBS datasets, in total of $155,384$ paired proteins with their surface features for retrieval. And the models are trained on all training PPIs.

In Tab. 8, we show the cross-dataset performance. We take the model trained on PDB training set only to encode the PPIs in DIPS and PPBS training and validation sets, respectively. Then we evaluate the cross-dataset performance on DIPS and PPBS test sets at two different hit rates, respectively. **Top10 hit** means that we identify the 10 most similar surface representations to the query protein in the test set for inference. Then, PPIretrieval decodes between the query protein and the 10 potential binding partners associated with these similar proteins. The best binding partner for the query protein is then selected based on the highest *dockQ* score.

Additionally, we show the quality distribution of PPIs predicted by PPIretrieval in comparison with ground-truth and reference at different hit rates in Fig 8. For *dockQ* score [2], a score in the range (0, 0.23) denotes incorrect interaction, [0.23, 0.49) denotes acceptable interaction, [0.49, 0.8) denotes medium interaction, and [0.8, 1) denotes good interaction. In Fig 8, we divide these ranges into sub-ranges for better visualization of the quality distribution, (0, 0.23)→(0, 0.1)∪[0.1, 0.23),

Dataset	Metrics		PDB	DIPS	PPBS
PPI Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4596	0.4797	0.4949
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.4039}$	$\underline{0.4042}$	$\underline{0.4092}$
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.3907	0.3935	0.4040
	TM(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4552	0.5909	0.5767
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.2196}$	0.4211	$\underline{0.3167}$
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.1950	$\underline{0.4346}$	0.3105
	rmsd(↓)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	7.38	6.70	6.70
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	11.60	$\underline{8.99}$	11.07
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	$\underline{10.80}$	10.34	$\underline{10.27}$
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.4220}$	$\underline{0.4304}$	$\underline{0.5946}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4073	0.4177	0.5535
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.2366}$	$\underline{0.6649}$	$\underline{0.4735}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.2134	0.6617	0.4622
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{10.44}$	$\underline{6.02}$	9.77
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	11.40	11.33	$\underline{8.20}$

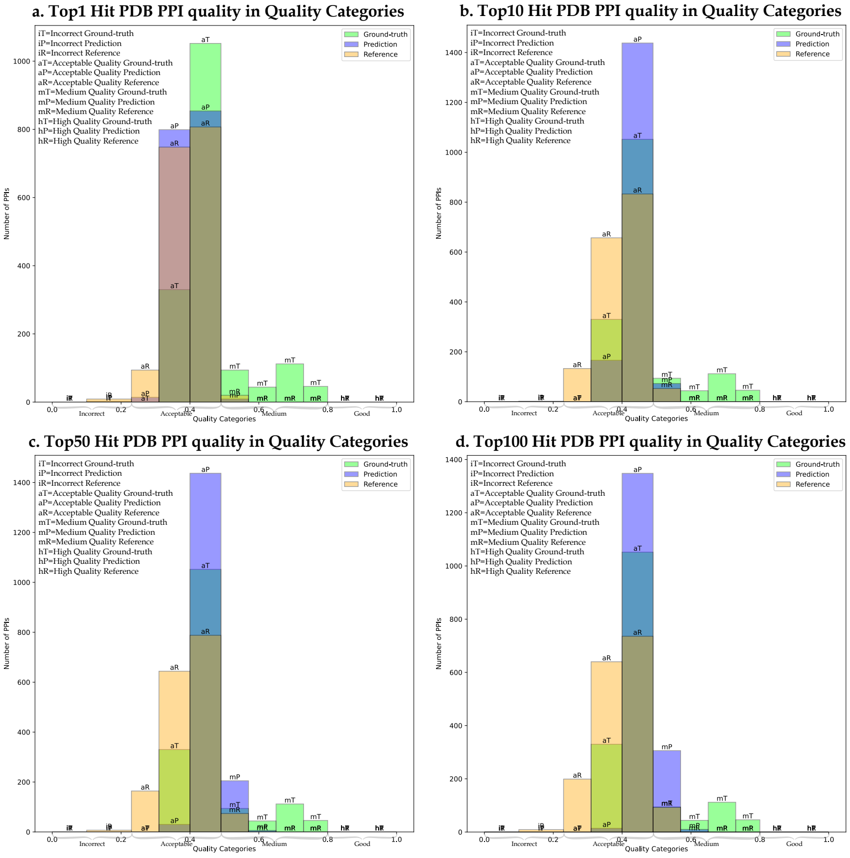
Table 6: *dockQ*, *TM*, and *rmsd* for evaluation of PPIs and binding sites of **Top1 hit** predicted by PPIretrieval in comparison with ground-truth structures over three runs. The database for each test set comprises surface representations from the training and validation sets of each respective dataset.

Dataset	Metrics		PDB	DIPS	PPBS
PPI Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4596	0.4797	0.4949
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.4110}$	$\underline{0.4394}$	$\underline{0.4400}$
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.4061	0.4093	0.4130
	TM(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4552	0.5909	0.5767
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.2649}$	$\underline{0.4507}$	$\underline{0.4148}$
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.2507	0.4394	0.4009
	rmsd(↓)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	7.38	6.70	6.70
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	10.78	$\underline{8.84}$	$\underline{9.11}$
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	$\underline{9.88}$	10.09	11.01
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.4678}$	$\underline{0.4410}$	$\underline{0.6045}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4250	0.4367	0.6014
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	$\underline{0.3300}$	$\underline{0.6944}$	$\underline{0.6045}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.3222	0.6914	0.6014
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	10.70	$\underline{5.67}$	$\underline{6.52}$
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	$\underline{9.30}$	9.65	9.96

Table 7: *dockQ*, *TM*, and *rmsd* for evaluation of PPIs and binding sites of **Top1 hit** predicted by PPIretrieval in comparison with ground-truth structures over three runs. The database comprises all surface representations from the training and validation sets of PDB, DIPS, and PPBS datasets.

Dataset	Metrics		DIPS-Top1	DIPS-Top10	PPBS-Top1	PPBS-Top10
PPI Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4797	0.4797	0.4949	0.4949
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{pred}}$	0.4334	0.4432	0.4098	0.4447
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{ref}}$	0.4054	0.4316	0.4040	0.4051
	TM(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.5909	0.5909	0.5767	0.5767
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{pred}}$	0.3965	0.4691	0.3219	0.3021
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{ref}}$	0.3708	0.4357	0.2903	0.2780
	rmsd(↓)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	6.70	6.70	6.70	6.70
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{pred}}$	8.71	10.49	12.09	10.25
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{ref}}$	11.68	9.26	10.71	10.13
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{pred}}$	0.4207	0.4435	0.5579	0.5857
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{ref}}$	0.4030	0.4156	0.5231	0.5611
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{pred}}$	0.5419	0.6792	0.4421	0.3889
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{ref}}$	0.5330	0.6714	0.4202	0.3725
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{pred}}$	5.84	10.50	11.76	10.49
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{ref}}$	11.12	7.35	8.92	8.91

Table 8: *dockQ*, *TM*, and *rmsd* for evaluation of PPIs and binding sites of **Top1, Top10 hit** predicted by PPIretrieval in comparison with ground-truth structures on cross-datasets over three runs. The database for each test set comprises surface representations from the training and validation sets of each respective dataset.


a. Top1 Hit PDB PPI quality in Quality Categories

b. Top10 Hit PDB PPI quality in Quality Categories

c. Top50 Hit PDB PPI quality in Quality Categories

d. Top100 Hit PDB PPI quality in Quality Categories

Figure 8: Comparison of PPI qualities in the test set of PDB dataset, considering ground-truth, predicted, and reference PPIs, evaluated using the *dockQ* score at **Top1, Top10, Top50, Top100 hit**. The database comprises surface representations from training and validation sets from PDB dataset only.

[0.23, 0.49)→[0.23, 0.31)∪[0.31, 0.4)∪[0.4, 0.49), [0.49, 0.8)→[0.49, 0.57)∪[0.57, 0.65)∪[0.65, 0.73)∪[0.65, 0.73)∪[0.65, 0.8),[0.8, 1)→[0.8, 0.9)∪[0.9, 1).

We observe that more predicted PPIs fall into the medium-quality category as hit rates increase from **Top1** to **Top100**, surpassing the number of ground-truth PPIs of acceptable and medium qualities.

This consistent quality distribution of PPIs predicted by PPIretrieval indicates a strong potential for novel PPI findings. Furthermore, we compare the results of using cosine similarity in Tab. 11 and Fig. 9.

C.1 More Ablation Results

In Tab. 9, we present experimental results for PPIretrieval at five different hit rates, increasing from **Top1** to **Top100**. The models are trained on PDB training set only.

PDB Dataset	Metrics		Top1	Top10	Top20	Top50	Top100
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4073	0.4362	0.4379	0.4411	0.3507
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.4220	0.4375	0.4459	0.4569	0.4688
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.1334	0.1338	0.1355	0.1401	0.1405
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.1155	0.1194	0.1212	0.1247	0.1301
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.2134	0.2078	0.2059	0.2059	0.2108
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.2366	0.2266	0.2241	0.2231	0.2265
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.0773	0.0775	0.0774	0.0758	0.0702
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.0665	0.0668	0.0679	0.0698	0.0701
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	11.40	9.74	9.59	9.50	9.34
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	10.44	9.33	8.94	8.52	8.16
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	15.98	15.88	15.84	15.76	15.53
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	17.87	17.31	17.03	16.55	16.02
Cost	*PPIretrieval runtime*(↓)	second/protein	0.29	0.91	1.97	4.64	9.44

Table 9: *dockQ*, *TM*, and *rmsd* for evaluation of **Top1, Top10, Top20, Top50, Top100 hit** binding sites predicted by PPIretrieval in comparison with other binding sites in the PDB test set over three runs. The database comprises surface features from training and validation sets from PDB dataset only.

We observe an improvement in the quality of predicted PPIs, measured by *dockQ, TM* scores and *rmsd*, as the hit rate increases from **Top1** to **Top100**. Notably, with 100 similar representations, the predicted PPIs exhibit high quality in terms of the *dockQ* and *TM* scores. This suggests potential PPI exploration with PPIretrieval.

Furthermore, we present experimental results for PPIretrieval at five different hit rates, increasing from **Top1** to **Top100**, in Tab. 10. The models are trained on all PDB, DIPS, and PPBS training set, and the database comprises surface representations from training and validation sets of them.

PDB Dataset	Metrics		Top1	Top10	Top20	Top50	Top100
Site Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.4126	0.4331	0.4480	0.4491	0.4490
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.4235	0.4402	0.4531	0.4649	0.4708
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.1433	0.1436	0.1455	0.1458	0.1478
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.1225	0.1266	0.1301	0.1398	0.1405
	TM(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.3944	0.3877	0.3833	0.3554	0.3422
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.4041	0.3969	0.3863	0.3625	0.3528
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	0.0787	0.0766	0.0750	0.0721	0.0709
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	0.0536	0.0548	0.0588	0.0582	0.0601
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	10.41	9.73	9.70	9.49	9.32
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	10.04	8.97	8.66	8.20	7.35
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{masif}}$	15.73	15.71	15.54	15.26	15.19
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{dmasif}}$	17.75	17.22	17.02	16.40	16.11

Table 10: *dockQ*, *TM*, and *rmsd* for evaluation of **Top1, Top10, Top20, Top50, Top100 hit** binding sites predicted by PPIretrieval in comparison with other binding sites in the PDB test set over three runs. The database comprises surface features from training and validation sets from PDB, DIPS, and PPBS dataset.

We observe improved predicted interface quality in terms of *dockQ, TM* scores, and *rmsd*, with larger surface database. The robust experimental results suggest that PPIretrieval has the potential to facilitate and expedite the discovery of novel PPIs, identifying candidates with higher *dockQ* scores. However, it is important to note that the computational time has also increased significantly. As a retrieval model, there exists a trade-off between performance and efficiency, and this trade-off becomes evident with higher hit rates.

D Additional Experiments on Euclidean Distance vs. Cosine Similarity

We show additional results and visualize more quality distributions carried out by PPIretrieval using Euclidean distance and cosine similarity in this section.

In Sec. A.3 and Fig. 1, we present the PPIretrieval inference strategy to find a binding partner for a query protein. We identify a similar surface representation to the query protein in the PPIretrieval database using the Euclidean distance function for the surface embeddings,

$$d(\mathbf{H}_P, \mathbf{H}_A) = \sqrt{\sum_{i=1}^{d}\left(\frac{1}{N}\sum_{j=1}^{N}\mathbf{H}_P^{j,i} - \frac{1}{M}\sum_{k=1}^{M}\mathbf{H}_A^{k,i}\right)^2} \in \mathbb{R}. \tag{7}$$

Here, $\mathbf{H}_P \in \mathbb{R}^{N\times d}$ represents the surface embedding of the query protein, and $\mathbf{H}_A \in \mathbb{R}^{M\times d}$ represents the surface embedding of a protein in our PPIretrieval database. In addition to the Euclidean distance, we can use cosine similarity to find the surface representation that is the most similar to the query protein.

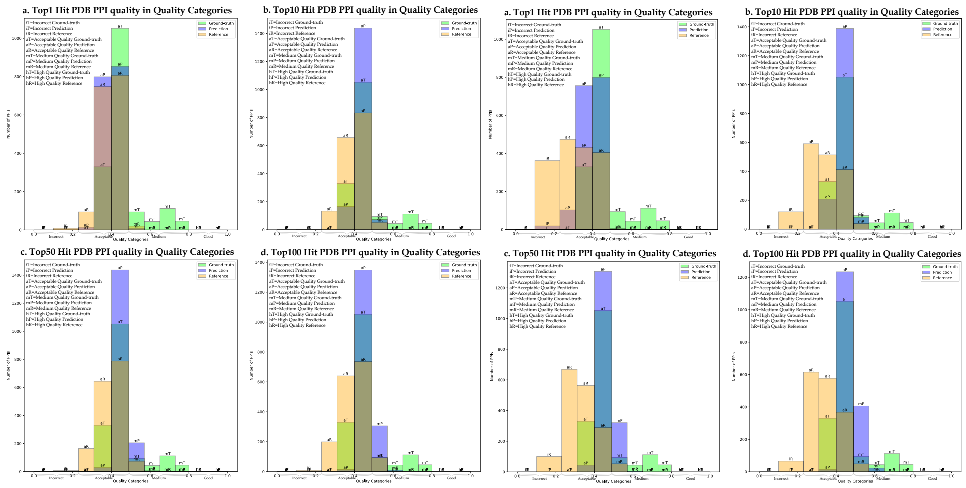
In Tab. 11, we present experimental results for PPIretrieval using Euclidean Distance and Cosine Similarity at five different hit rates, ranging from **Top1** to **Top100**. The models are exclusively trained on the PDB training set. Our observations reveal an improvement in the quality of predicted PPIs, as measured by the *dockQ* score and *rmsd*, when employing Cosine Similarity to find similar surface representations compared to using Euclidean Distance for the same purpose in the PPIretrieval database. During the inference stage, we leverage Cosine Similarity as an alternative method for retrieving similar surface representations for the query protein, in contrast to the original approach using Euclidean distance.

PDB Dataset	Metrics		Top1	Top10	Top20	Top50	Top100
PPI Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4596	0.4596	0.4596	0.4596	0.4596
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	0.4039	0.4352	0.4428	0.4532	0.4614
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.3907	0.3920	0.3943	0.3963	0.3968
	TM(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4552	0.4552	0.4552	0.4552	0.4552
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	0.2196	0.2183	0.0.2174	0.2168	0.2159
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.1950	0.1938	0.1944	0.1950	0.1958
	rmsd(↓)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	7.38	7.38	7.38	7.38	7.38
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	11.60	9.52	9.17	8.73	8.42
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	10.80	9.77	9.65	9.65	9.56
Site Quality	*TM*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.2366	0.2195	0.2174	0.2154	0.2156
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.2134	0.1986	0.1960	0.1955	0.1957
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	11.52	10.01	9.84	9.62	9.42
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	10.50	9.69	9.62	9.53	9.47

PDB Dataset	Metrics		Top1	Top10	Top20	Top50	Top100
PPI Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4596	0.4596	0.4596	0.4596	0.4596
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	0.3907	0.4381	0.4472	0.4601	0.4683
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.3147	0.3319	0.3388	0.3407	0.3422
	TM(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4552	0.4552	0.4552	0.4552	0.4552
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	0.2009	0.2101	0.2085	0.2089	0.2095
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	0.1838	0.1857	0.1861	0.1934	0.1941
	rmsd(↓)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	7.38	7.38	7.38	7.38	7.38
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{pred}}$	11.08	9.15	8.86	8.39	7.99
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_B^{\text{ref}}$	11.01	9.91	9.81	9.92	9.79
Site Quality	*TM*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	0.1981	0.2084	0.2082	0.2049	0.2055
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	0.1818	0.1832	0.1860	0.1923	0.1919
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{pred}}$	11.05	9.56	9.42	9.13	9.09
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_B^{\text{ref}}$	10.90	9.93	9.83	9.80	9.87

Table 11: Left: Retrieval Using Euclidean Distance. Right: Retrieval Using Consine Similarity.*dockQ*, *TM*, and *rmsd* for evaluation of PPIs and binding sites of **Top1, Top10, Top20, Top50, Top100 hit** predicted by PPIretrieval in comparison with ground-truth structures on PDB dataset over three runs. The database comprises surface representations from training and validation sets from PDB dataset only.

In addition to the tabular results in Tab. 11, we visualize the quality distribution of predicted PPIs by PPIretrieval using Euclidean Distance and Cosine Similarity in Fig. 9. Our observations reveal that, as hit rates increase from **Top1** to **Top100**, more predicted PPIs fall into the medium-quality category when using cosine similarity to retrieve surface representations similar to the query protein. This trend exceeds the number of ground-truth PPIs of acceptable and medium qualities. The consistent quality distribution of predicted PPIs by PPIretrieval suggests a robust potential for novel PPI exploration with PPIretrieval. Additionally, we provide Cosine Similarity as an alternative choice to the Euclidean distance in our approach.



(a) Retrieval Using Euclidean Distance

(b) Retrieval Using Cosine Similarity

Figure 9: Comparison of PPI qualities in the test set of PDB dataset, considering ground-truth, predicted, and reference PPIs, evaluated using the *dockQ* score at **Top1, Top10, Top50, Top100 hit**. The database comprises surface representations from training and validation sets from PDB dataset only.

E Additional Experiments on Cross-Dataset Validation

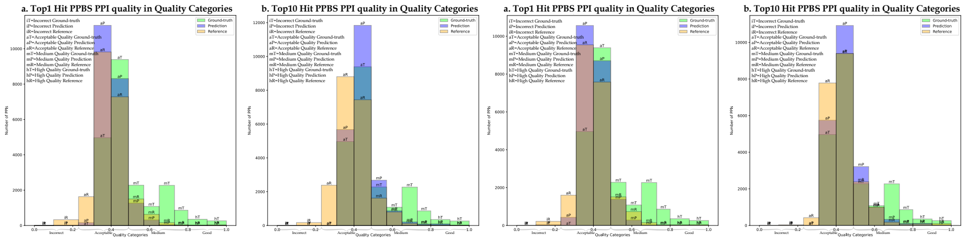
We show additional results and visualize more quality distributions carried out by PPIretrieval on the cross-dataset validation. Following the cross-dataset validation results in Tab. 2, we provide a thorough experimental analysis and visualization here.

In Table 12, we present the cross-dataset validation results for **Top1** and **Top10 hits** on the PPBS test set. We utilize two models for this analysis: one is trained on the PPBS training set and validated on the PPBS test set, and the other is trained on the PDB training set and *cross-validated* on the PPBS test set. With **Top1 hit**, we find that PPIretrieval is capable of generalizing to unseen protein complexes, as evidenced by the second model exhibiting better PPI quality compared to the first model in terms of *dockQ* and *TM* scores. When we expand our retrieval space to **Top10 hit**, the first model (trained on the PPBS training set) predicts PPIs with improved quality.

Dataset	Metrics		PPBS-Top1	PPBS-Top1 (PDB Cross-Dataset)	PPBS-Top10	PPBS-Top10 (PDB Cross-Dataset)
PPI Quality	*dockQ*(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.4949	0.4949	0.4949	0.4949
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{pred}}$	0.4092	<u>0.4098</u>	<u>0.4513</u>	0.4447
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{ref}}$	0.4040	0.4040	0.4345	0.4051
	TM(↑)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	0.5767	0.5767	0.5767	0.5767
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{pred}}$	0.3167	<u>0.3219</u>	<u>0.3245</u>	0.3021
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{ref}}$	0.3105	0.2903	0.3219	0.2780
	rmsd(↓)	$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{\text{true}}$	6.70	6.70	6.70	6.70
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{pred}}$	11.07	12.09	9.46	10.25
		$\mathbf{Y}_{\text{query}}, \mathbf{Y}_{B}^{\text{ref}}$	10.27	10.71	9.53	10.13
Site Quality	*TM*(↑)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{pred}}$	0.4652	0.4421	0.4323	0.3889
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{ref}}$	0.4747	0.4202	0.4396	0.3725
	rmsd(↓)	$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{pred}}$	9.77	11.76	8.95	10.49
		$\mathbf{Y}_{\text{true}}, \mathbf{Y}_{B}^{\text{ref}}$	8.20	8.92	8.15	8.91

Table 12: *dockQ*, *TM*, and *rmsd* for evaluation of PPIs and binding sites of **Top1, Top10 hit** predicted by PPIretrieval in comparison with ground-truth structures on cross-datasets over three runs. The database for each test set comprises surface representations from the training and validation sets of each respective dataset.

In addition to the results presented in Table 12, we also provide a visualization of the quality distribution of predicted PPIs by PPIretrieval for the cross-dataset validation in Fig. 10. Fig. 10(a) displays the results of the model trained on the PDB training set and then cross-validated on the PPBS test set, while Fig. 10(b) shows the results of the model trained on the PPBS training set and then validated on the PPBS test set. With **Top1 hit**, it is noticeable that both models predict PPIs of approximately the same quality, as evidenced by comparing the quality distributions. This visual



(a) PDB Model Cross-Validated on PPBS Test Set (b) PPBS Model Validated on PPBS Test Set

Figure 10: Comparison of PPI qualities in the test set of PPBS dataset, considering ground-truth, predicted, and reference PPIs, evaluated using the *dockQ* score at **Top1, Top10 hit**. (a) presents results from the model trained on the PDB training set and cross-validated on the PPBS test set. (b) presents results from the model trained on the PPBS training set and validated on the PPBS test set.

representation demonstrates that PPIretrieval possesses the capability to generalize to unseen protein complexes and accurately predict their interactions.

F Additional Experiments on Binding Interface Prediction

We show additional results on the binding interface prediction task.

We evaluate the models on PDB and PPBS datasets (details reported in Sec. 7). For baseline comparisons, we train MaSIF-search [9] and dMaSIF-search[2] [35] to predict the binding interfaces of receptor and ligand proteins in a complex. The comparisons are summarized and highlighted in Tab. 13. MaSIF [9] and dMaSIF [35] offer model variants, namely MaSIF-search and dMaSIF-search, for predicting interactions between protein complexes. To enable these models to make predictions for the binding interfaces of protein complexes, we extract their final embeddings, denoted as $\mathbf{F}_R \in \mathbb{R}^{M_R \times d}$ and $\mathbf{F}_L \in \mathbb{R}^{M_L \times d}$, just before the output layer. Employing the strategy outlined in Sec. 3.2 to predict the surface-level binding interface, we employ a MLP with sigmoid function directly on these embeddings, resulting in $\hat{\mathbf{Y}}_R^{\text{surf}} \leftarrow \sigma(\text{MLP}(\mathbf{F}_R)) \in [0,1]^{M_R \times 1}$, $\hat{\mathbf{Y}}_L^{\text{surf}} \leftarrow \sigma(\text{MLP}(\mathbf{F}_L)) \in [0,1]^{M_L \times 1}$. We then compute the residue-level binding interfaces, denoted as $\hat{\mathbf{Y}}_R^{\text{res}} \in \{0,1\}^{N_R \times 1}$, $\hat{\mathbf{Y}}_L^{\text{res}} \in \{0,1\}^{N_L \times 1}$ from $\hat{\mathbf{Y}}_R^{\text{surf}}$, $\hat{\mathbf{Y}}_L^{\text{surf}}$, respectively. For each residue i in R, we define a region with a fixed radius of $r = 10\mathring{A}$ and collect a set of surface points within this region, each with a binding interface $\hat{\mathbf{y}}_j^{\text{surf}}$ and embedding $\mathbf{f}_j$. The residue i is considered part of the binding interface if the majority of surface points in the region are labeled as part of the binding interface, i.e., $\hat{\mathbf{y}}_i^{\text{res}} = 1$ if $\text{Mean}(\sum_j \hat{\mathbf{y}}_j^{\text{surf}}) > 0.5$; otherwise $\hat{\mathbf{y}}_i^{\text{res}} = 0$. The residue-level binding interface $\hat{\mathbf{Y}}_L^{\text{res}}$ for L is computed by the same method. In terms of training strategy, these baseline models are directly optimized through the binding interface optimization in Sec. 4.3.

Dataset	PDB		PPBS	
Model	**Acc(↑)**	**ROC(↑)**	**Acc(↑)**	**ROC(↑)**
MaSIF-search	22.93	20.10	20.45	19.79
dMaSIF-search	20.86	20.74	22.35	21.08
PPIretrieval	**92.76**	**92.61**	**93.55**	**94.98**

Table 13: Accuracy and ROC of PPIretrieval in comparison with MaSIF-search and dMaSIF-search on datasets over 5 runs.

In Table 13, we observe that PPIretrieval surpasses MaSIF-search and dMaSIF-search, achieving improvements of $69.83\%, 71.90\%$ in accuracy, and $72.51\%, 71.87\%$ in ROC on the PDB dataset. Additionally, PPIretrieval outperforms MaSIF-search and dMaSIF-search by $73.10\%, 71.20\%$ in accuracy, and $75.19\%, 73.90\%$ in ROC on the PPBS dataset.

[2] The MaSIF and dMaSIF models were originally designed without prediction capabilities for binding interfaces of protein complexes. We have modified their architectures to enable the models to make predictions for binding interfaces.

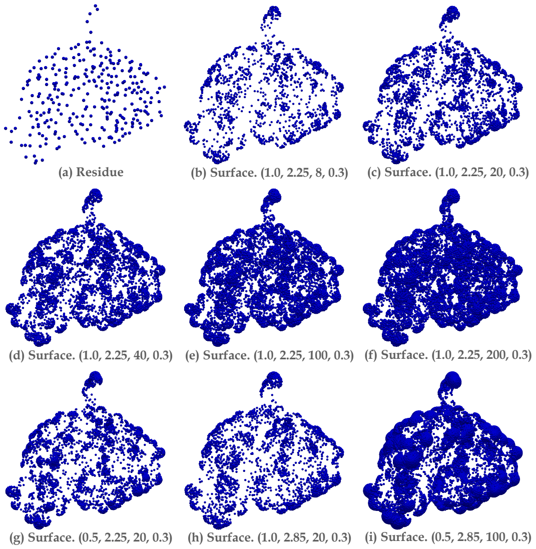
Abalation Study In Tab. 14, we conduct an ablation study to examine the effectiveness of our training objectives for the PPIretrieval model, which includes the *lock-and-key* goal $\mathcal{L}_{\text{match}}$, the contrastive goal $\mathcal{L}_{\text{contra}}$, and the binding interface goal $\mathcal{L}_{\text{bind}}$.

Dataset	$\mathcal{L}_{\text{match}}$	$\mathcal{L}_{\text{contra}}$	$\mathcal{L}_{\text{bind}}$	**Acc**(↑)	**ROC**(↑)
PDB			✓	91.10	90.88
PDB		✓	✓	92.11	91.13
PDB	✓		✓	90.86	91.95
PDB	✓	✓	✓	**92.76**	**92.61**

Table 14: Ablation study of PPIretrieval optimization. A checkmark (✓) indicates that an objective is used to optimize the model.

We observe that, overall, PPIretrieval performs better in terms of both accuracy and ROC when all three optimization objectives are combined. This suggests that PPIretrieval successfully learns the *lock-and-key* structure between the receptor and ligand in a protein complex under our training strategy.

In Fig. 11, we visualize protein surface sampling with different dMaSIF parameters. We can observe that lower sampling resolution, higher sampling distance, and a greater number of sampling points contribute to a more accurate approximation of the protein surface manifold. However, due to limited computational resources, we can only perform sampling and training shown in Fig. 11(c). Training PPIretrieval on more precise surface sampling with more computing resources is a potential direction for exploration.



(a) Residue (b) Surface. (1.0, 2.25, 8, 0.3) (c) Surface. (1.0, 2.25, 20, 0.3)

(d) Surface. (1.0, 2.25, 40, 0.3) (e) Surface. (1.0, 2.25, 100, 0.3) (f) Surface. (1.0, 2.25, 200, 0.3)

(g) Surface. (0.5, 2.25, 20, 0.3) (h) Surface. (1.0, 2.85, 20, 0.3) (i) Surface. (0.5, 2.85, 100, 0.3)

Figure 11: Visualization of dMaSIF parameters for surface sampling. The bracket includes (Resolution, Distance, #Sample, Variance).

In Tab. 15, we present the performance of PPIretrieval with different dMaSIF sampling parameters. PPIretrieval exhibits improved accuracy and ROC on PDB, with more precise protein surface sampling.

Dataset	Resolution	Distance	#Sample	Variance	#Surf Point	**Acc**(↑)	**ROC**(↑)
PDB	1.00	2.25	8	0.30	2175	90.16	90.96
PDB	1.00	2.25	12	0.30	2425	91.44	90.99
PDB	1.00	2.85	20	0.30	2632	92.03	91.55
PDB	1.00	2.25	20	0.30	2680	92.76	92.61

Table 15: Ablation study of dMaSIF parameters for PPIretrieval. *#Surf Point* denotes the average number of surface points of a protein sampled by dMaSIF across the training set.

Once again, these results motivate further exploration for future enhancements.

Computational Resources Our models are trained on a single Nvidia 48G A40 GPU. Regarding training time, dMaSIF-search takes approximately 0.29s to train a protein complex, while PPIretrieval takes around 0.35s for the same task. In terms of inference time, dMaSIF-search requires about 0.10s for a protein complex, while PPIretrieval takes approximately 0.11s.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Full experiments

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 8

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: NA

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Provided with code links for checking

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Full code access

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All discussed in Section 7

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bar NA for such task

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Single A40 GPU

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Well confirmed

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Well discussed

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Full credits

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.