UniSite: The First Cross-Structure Dataset and Learning Framework for End-to-End Ligand Binding Site Detection

Jigang Fan[1,*] **Quanlin Wu**[1,*] **Shengjie Luo**[2] **Liwei Wang**[1,2,3,†]

[1]Center for Data Science, Peking University
[2]State Key Laboratory of General Artificial Intelligence, Peking University
[3]Center for Machine Learning Research, Peking University
{jigangfan,luosj}@stu.pku.edu.cn, {quanlin,wanglw}@pku.edu.cn

Abstract

The detection of ligand binding sites for proteins is a fundamental step in Structure-Based Drug Design. Despite notable advances in recent years, existing methods, datasets, and evaluation metrics are confronted with several key challenges: (1) current datasets and methods are centered on individual protein–ligand complexes and neglect that diverse binding sites may exist across multiple complexes of the same protein, introducing significant statistical bias; (2) ligand binding site detection is typically modeled as a discontinuous workflow, employing binary segmentation and subsequent clustering algorithms; (3) traditional evaluation metrics do not adequately reflect the actual performance of different binding site prediction methods. To address these issues, we first introduce UniSite-DS, the first UniProt (Unique Protein)-centric ligand binding site dataset, which contains 4.81 times more multi-site data and 2.08 times more overall data compared to the previously most widely used datasets. We then propose UniSite, the first end-to-end ligand binding site detection framework supervised by set prediction loss with bijective matching. In addition, we introduce Average Precision based on Intersection over Union (IoU) as a more accurate evaluation metric for ligand binding site prediction. Extensive experiments on UniSite-DS and several representative benchmark datasets demonstrate that IoU-based Average Precision provides a more accurate reflection of prediction quality, and that UniSite outperforms current state-of-the-art methods in ligand binding site detection. The dataset and codes will be made publicly available at `https://github.com/quanlin-wu/unisite`.

1 Introduction

The detection of ligand binding sites on target proteins is one of the most critical steps in modern drug discovery strategies [1, 2, 3]. Structure-based drug design approaches begin with the three-dimensional structure of the target protein, from which deep, druggable cavities are identified. These regions, referred to as binding sites or binding pockets, are composed of sets of protein residues. Once the protein's sites are recognized, virtual screening of a molecular library can be performed using methods such as protein–ligand docking and protein–ligand affinity prediction [4, 5]. Alternatively, *de novo* molecular design [6, 7] can be conducted based on the local structure of the binding sites to identify potential candidate compounds. As a fundamental step, the accurate identification of protein binding sites can significantly facilitate and influence subsequent steps in drug discovery.

[*]Equal contribution.
[†]Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Over the past several decades, some endeavours have been made to detect protein ligand binding sites. These methods have evolved from traditional techniques based on geometry [8], template searching [9], and energy probes [10], to machine learning methods based on surface features [11], and further to deep learning methods utilizing convolutional neural networks (CNNs) [12] and graph neural networks (GNNs) [13, 14, 15]. Concurrently, a series of protein–ligand datasets have also been established progressively, including scPDB [16] and PDBbind [17] datasets for protein–ligand complex structures, as well as benchmark datasets such as HOLO4K [18] and COACH420 [19] for evaluating binding site detection methods.

Although the above efforts have significantly advanced the field of ligand binding site detection, current methods, datasets, and evaluation metrics are confronted with substantial challenges:

Issue 1. All previous methods and datasets are PDB (Protein Data Bank file)-centric, specifically focusing on individual protein–ligand structures, which introduces considerable statistical bias. Due to experimental constraints, only a limited number of binding sites in the protein are typically observed in one single protein–ligand structure where ligands are bound. However, one protein can be associated with numerous distinct protein–ligand structures, which exhibit high structural similarity in their protein components yet considerable variation in their binding site regions [20, 21, 22] (Figure 1). But existing datasets and methods only regard these structures as individual data entries, focusing on limited binding sites in single PDB structure. Training and evaluating on PDB-centric datasets introduces significant statistical bias, as the annotation paradigm of individual PDB structures overlooks many other ground truth binding sites.

Issue 2. Existing methods employ discontinuous workflows for binding site detection. Most approaches [8, 11, 13, 14] first perform semantic segmentation to generate binary masks of potential binding residues/atoms, then cluster them into discrete binding sites. Alternative implementations only predict binding site centers [15], and the associated residues need to be extracted using external methods. These fragmented pipelines highly rely on the post-processing methods (e.g. clustering algorithms), inherently limit end-to-end optimization and struggle with overlapping binding sites.

Issue 3. Traditional evaluation metrics inadequately reflect the actual performance of binding site detection. The most widely used evaluation metrics are DCC and DCA [11]. DCC represents the distance between the predicted binding site center and the ground truth binding site center. DCA denotes the shortest distance between the predicted binding site center and any heavy atom of the ligand. These metrics suffer from two fundamental limitations (Figure 4): (1) they completely disregard the structural properties such as shape, size, and residue composition of binding sites, which are crucial for downstream tasks (Appendix A), and (2) the absence of proper matching criteria between predictions and ground truth may lead to double-counting of predictions.

To address the issues mentioned above, this paper makes the following contributions:

1) We introduce **UniSite-DS**, a **manually curated, UniProt (Unique Protein)-centric** dataset of protein ligand binding sites. Leveraging the unique identifiers assigned to protein sequences in UniProt [23], we systematically integrated all ligand binding sites associated with given unique protein across multiple PDB structures. To the best of our knowledge, it is the first UniProt-centric dataset. Notably, UniSite-DS includes **4.81** times more multi-site proteins than existing datasets [16, 17], and the overall size of the dataset is **2.08** times larger. The Uniprot-centric dataset corrects the statistical bias of previous PDB-centric datasets, thereby resolving Issue 1 and significantly broadening the available data.

2) We propose **UniSite-1D** and **UniSite-3D**, two **end-to-end** methods for protein ligand binding site detection. Both models utilize a transformer encoder-decoder architecture, supervised by a set prediction loss with bijective matching. UniSite 1D/3D directly predict N potentially overlapping binding sites without requiring post-processing clustering steps, thus completely resolves Issue 2. The **UniSite-1D** variant operates exclusively on 1D protein sequence inputs, providing structure-free binding site detection capability. For enhanced performance, the **UniSite-3D** variant incorporates 3D structural information while maintaining the same end-to-end prediction framework.

3) To overcome the limitations inherent in traditional evaluation methods outlined in Issue 3, we introduce an **Average Precision (AP)** metric based on **Intersection over Union (IoU)** for fair and comprehensive binding site assessment. Extensive experiments have demonstrated that the IoU-based AP maintains strong concordance with method rankings under traditional metrics while overcoming their key limitations, providing a more accurate reflection of prediction quality.

4) Extensive experiments on UniSite-DS and classical datasets have demonstrated that our methods outperform the current state-of-the-art methods in protein ligand binding site detection. These results indicate that the end-to-end detection framework, which operates without the need for specialized feature engineering, is already capable of exhibiting strong performance for binding site detection.

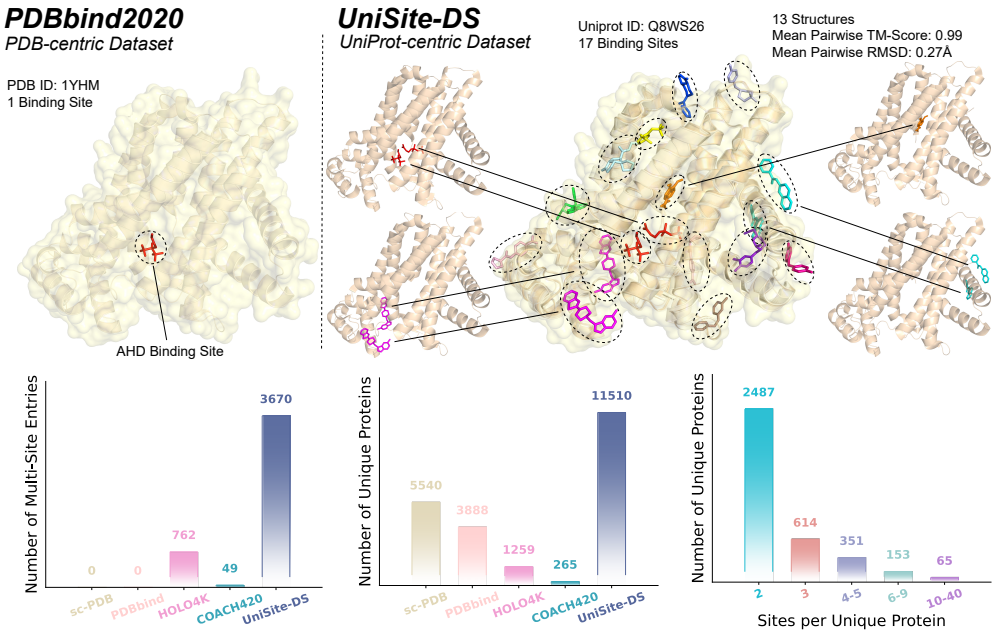

Figure 1: **Comparison between UniSite-DS and previous datasets. (Top left)** In PDBbind2020, only one ligand binding site and one structure are recorded for UniProt ID Q8WS26. **(Top right)** In contrast, UniSite-DS integrates distinct binding sites across all available structures (highly similar, mean TM-Score=0.99), identifying 17 unique ligand binding sites derived from 13 representative PDB entries. **(Bottom left and center)** Comparison of UniSite-DS with other widely used datasets in terms of multi-site entries and the number of unique proteins. For HOLO4K and COACH420, the most widely used *mlig* subsets were selected, where each entry corresponds to a PDB structure, while in UniSite-DS, each entry corresponds to a UniProt ID. **(Bottom right)** Distribution of the number of unique proteins in UniSite-DS with respect to the number of distinct binding sites they contain.

2 UniSite-DS: The First Uniprot-centric Dataset

A key challenge in detecting protein binding sites is how to identify all potential binding sites [20, 21]. Most proteins contain an inherently conserved binding site, commonly referred to as the active site. The active site is shared among members of the same protein family, which means that molecules targeting this site will simultaneously target all other proteins within the family, which is highly likely to lead to off-target effects and side effects [24]. Identifying other binding sites within the protein that can be targeted is a crucial strategy. These sites are often located in regions topologically distant from the active site and can modulate the protein's function through allosteric effects [25, 26, 27].

As illustrated in Figure 1, one single protein can correspond to a large number of different ligand-bound structures. While the overall protein structure tends to be highly conserved, the ligand binding site regions vary considerably across these structures. The motivation behind constructing UniSite-DS lies in the recognition that identifying all potential binding sites of a protein requires a comprehensive examination of all its ligand-bound structures—an important consideration that has been overlooked by previous methods and datasets.

To construct UniSite-DS, we performed the following search and processing steps: (1) We utilized AHoJ [28] to systematically search for all protein–ligand interactions in the PDB database [29]; (2) To ensure dataset quality, we excluded entries with a resolution greater than 2.5Å or those determined by non-crystallographic methods; (3) Following P2Rank's filtering criteria [11], we removed solvent molecules and ligands composed of fewer than five atoms, resulting in a total

of 143,197 protein–ligand interaction entries; (4) For each interaction, binding site residues were identified within a 4.5Å radius of the ligand; (5) We discarded entries with three or fewer binding site residues to eliminate "floating" ligands; (6) Leveraging UniProt's unique protein sequence identifiers [23], we mapped binding site residues from all protein–ligand interactions of each UniProt entry to their corresponding sequences via SIFTS annotations [30], integrating all ligand binding sites across different PDB structures; (7) To eliminate data redundancy among ligand binding sites, we applied Non-Maximum Suppression (NMS) with an Intersection over Minimum (IoM) threshold of 0.7 and an Intersection over Union (IoU) threshold of 0.5, excluding highly overlapping sites. This process resulted in 13,464 distinct UniProt IDs, of which 4,846 contained multiple ligand-binding sites; (8) Based on the criteria from Proteina [31], we set the sequence length threshold to 800; (9) **We manually inspected all UniProt IDs with more than ten ligand binding sites, as well as those where a single protein–ligand complex structure contributed three or more binding sites.** As a result, we identified 11,510 valid UniProt IDs, including 3,670 with multiple ligand binding sites. The distribution of ligand binding sites is shown in Figure 1. More details about the UniSite-DS curation workflow and manual inspection process are provided in Appendix B.

As the first UniProt-centric dataset, UniSite-DS encompasses **4.81** times more multi-site entries than previous datasets, and covers **2.96** times more UniProt entries than the widely used PDBbind dataset [17], as well as **2.08** times more than sc-PDB [16] (Figure 1). UniSite-DS eliminates the statistical biases inherent in earlier datasets and significantly expands the available data on multi-site ligand binding sites. Notably, case studies conducted using UniSite-DS (Appendix E) highlighted the limitations of current binding site prediction methods in handling multi-site proteins. This observation motivated us to develop a novel end-to-end method for protein–ligand binding site detection.

3 The Proposed Methodology

In this paper, we formulate protein ligand binding site detection as a **set prediction** task: given a protein P with an amino acid sequence S of length L, the goal of binding site detection is to identify **a set of binding sites** $\{m_i^{gt}\}_{i=1}^{N_{gt}}$, where each binding site is represented by a binary mask $m_i^{gt} \in \{0,1\}^L$. Here, $m_{ij}^{gt} = 1$ indicates that the j-th residue is part of the i-th site, while $m_{ij}^{gt} = 0$ means it is not. Currently, most learning-based binding site detection methods adopt a discontinuous workflow: first predicting a score for each amino acid residue or heavy atom, and then clustering them into distinct binding sites. To streamline this process, we propose UniSite, the first UniProt-centric and direct set prediction approach that adheres to the end-to-end paradigm (Figure 2). Two components are essential for direct set prediction in this context: (1) a set prediction loss based on bijective matching between predicted and ground truth binding sites; and (2) an architecture capable of predicting a set of sites in a single forward pass. The architecture of UniSite is shown in detail in Figure 3.

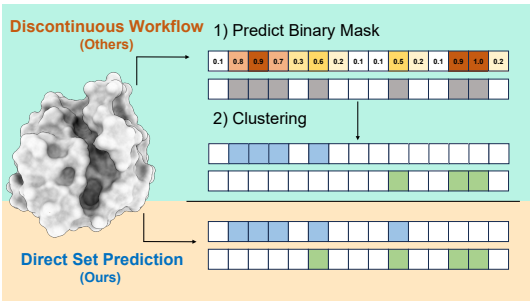

Figure 2: **Comparison of detection approaches. (Top)** Conventional learning-based binding site detection methods typically employ a discontinuous workflow: first predicting binary masks for residues/atoms, then clustering these masks into distinct binding sites. **(Bottom)** In contrast, our method directly outputs a set of N potentially overlapping binding sites in a single step.

3.1 Set prediction loss for binding site detection

UniSite infers a fixed-size set of N predictions $z = \{(p_i, m_i)|m_i \in \{0,1\}^L\}_{i=1}^N$ in a single forward pass, where m_i represents the predicted binding site, and p_i denotes the probability of binding and $\emptyset$ (non-binding) category. Since the ground truth set $|z^{gt}| = N^{gt}$ and the prediction set $|z| = N$ typically have unequal sizes, we assume $N \geq N^{gt}$ and pad the ground truth set with $\emptyset$ (non-binding) tokens. The padded ground truth set is defined as $z_{pad}^{gt} = \{(c_i^{gt}, m_i^{gt})|c_i^{gt} \in \{1, \emptyset\}, m_i^{gt} \in \{0,1\}^L\}_{i=1}^N$, where $c_i^{gt} = 1$ indicates a true binding site and $c_i^{gt} = \emptyset$ corresponds to the padding.

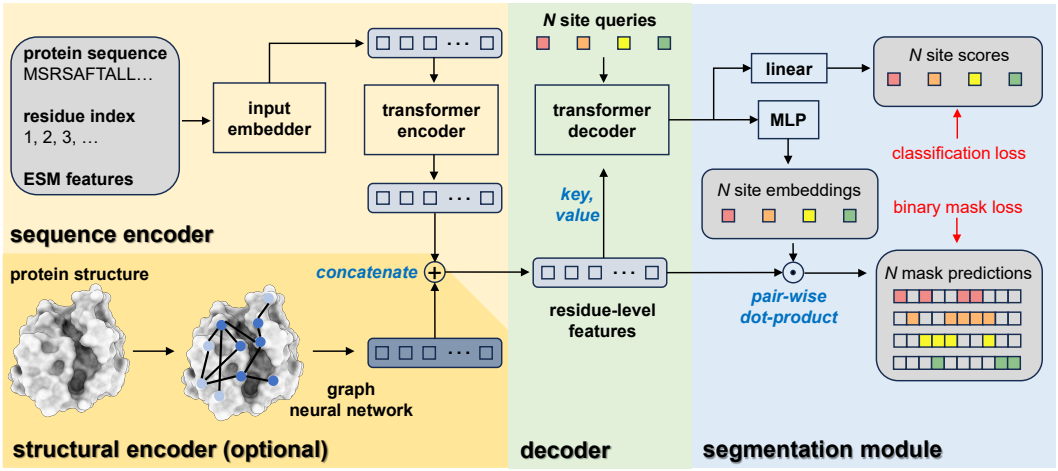

Figure 3: **The architecture of UniSite.** Our models employ an encoder to extract the residue-level features. Then a decoder module is used to generate embeddings of the N predicted binding sites. Finally, the segmentation module outputs N potentially overlapping binding site predictions. The encoder comprises dual pathways: a sequence encoder and an optional structural encoder, allowing UniSite to operate with either sequence-only input or combined sequence-structure information.

To train a set prediction model, we require a bijective matching σ between the predicted set z and the padded ground truth set z_{pad}^{gt}. This matching is obtained by minimizing matching cost $\mathcal{L}_{\text{match}}$:

$$\hat{\sigma} = \arg\min \sum_{i}^{N} \mathcal{L}_{\text{match}}(z_i^{gt}, z_{\sigma(i)}) \tag{1}$$

where σ is a permutation of N elements and $\mathcal{L}_{\text{match}}$ quantifies the pairwise matching cost between ground truth site z_i^{gt} and the prediction with index $\sigma(i)$. Following prior work [32, 33], we employ the Hungarian algorithm to compute the optimal matching and use the matching cost defined as:

$$\mathcal{L}_{\text{match}}(z_i^{gt}, z_{\sigma(i)}) = -\mathbf{1}_{\{c_i^{gt} \neq \emptyset\}} \log p_{\sigma(i)}(c_i^{gt}) + \mathbf{1}_{\{c_i^{gt} \neq \emptyset\}} \mathcal{L}_{\text{mask}}(m_i^{gt}, m_{\sigma(i)}) \tag{2}$$

where $\mathcal{L}_{\text{mask}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}$ is a combination of BCE loss and dice loss [34], $p_{\sigma(i)}$ and $m_{\sigma(i)}$ denote the predicted probability and the binding site for the $\sigma(i)$-th prediction, respectively. This matching cost considers both the class prediction and the similarity of the predicted and ground truth binding sites. Given the optimal matching $\hat{\sigma}$, we compose a cross-entropy classification loss and the binary mask loss $\mathcal{L}_{\text{mask}}$ for each predicted site to train model parameters:

$$\mathcal{L}_{\text{mask\&cls}}(z^{gt}, z) = \lambda_{\text{cls}} \sum_{i}^{N} -\log p_{\hat{\sigma}(i)}(c_i^{gt}) + \mathbf{1}_{\{c_i^{gt} \neq \emptyset\}} \mathcal{L}_{\text{mask}}(m_i^{gt}, m_{\hat{\sigma}(i)}) \tag{3}$$

3.2 UniSite architecture

As illustrated in Figure 3, UniSite comprises three main components: (1) an encoder module that extracts residue-level representations $\mathcal{F} \in \mathbb{R}^{L \times d_{\text{model}}}$ of the protein; (2) a decoder module consisting of multiple Transformer decoder layers to generate embeddings of the N predicted binding sites, and (3) a segmentation module which combines the residue-level representations and the decoder embeddings to produce the final predictions $\{(p_i, m_i) | m_i \in \{0,1\}^L\}_{i=1}^N$. This architecture maintains conciseness while demonstrating strong compatibility with existing protein representation methods. In our implementation, we construct two variants: UniSite-1D using only sequence encoding, and UniSite-3D incorporating both sequence and structural encoders.

Sequence encoder. The amino acid sequence encodes the primary information of a protein and serves as the fundamental input for the UniProt-centric binding site detection. First, an input embedding module receives three inputs: (1) learnable embeddings for the 21 amino acid types (20 standard amino acids plus an "unknown" category); (2) sinusoidal positional embedding [35] for residue indices; (3) pre-trained ESM-2 [36] protein embeddings. These three components are

concatenated along the feature dimension, and subsequently processed by a 3-layer multilayer perceptron (MLP) to generate the initial per-residue features. Then a stack of Transformer encoder layers process the combined features to capture residue-residue interactions and global sequence patterns.

Structural encoder. Protein structure serves as the most critical input for structure-based tasks, including ligand binding site detection. Recent advances have proposed various structural feature extraction approaches, including hand-crafted algorithms [11], CNN-based methods [12, 37] and graph neural network approaches [13, 15]. To demonstrate the generality and effectiveness of our approach, we utilize GearNet-Edge [38], a standard E(3)-invariant GNN model, without introducing any custom architecture or specialized feature engineering.

Given a protein P, the protein structure is represented as a residue-level relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ and $\mathcal{E}$ represent the set of nodes and edges respectively, and $\mathcal{R}$ is the set of edge types. Each node in the protein graph represents the alpha carbon of a residue, while sequential edges, radius edges and K-nearest neighbor edges are considered in the graph. Based on the defined protein graph, the node features are updated through the relational graph convolution layers [39] as follows:

$$u_i^{(l)} = \text{ReLU}\left(\text{BN}\left(\sum_{r\in\mathcal{R}} W_r \sum_{j\in\mathcal{N}_r(i)} h_j^{(l-1)}\right)\right), \; h_i^{(l)} = h_i^{(l-1)} + u_i^{(l)} \tag{4}$$

where $h_i^{(l)}$ represents the feature of node i at the l-th layer, $\mathcal{N}_r(i) = \{j \in \mathcal{V} | (j, i, r) \in \mathcal{E}\}$ denotes the neighborhood of node i with the edge type r, and W_r is the convolutional kernel matrix shared within the edge type r. Specifically, BN represents a batch normalization layer and ReLU denotes the ReLU activation function.

Unlike conventional binding site detection methods that rely on structural input, our framework allows the structural encoder to be optionally included. When incorporated, the structural features are concatenated with sequence features and projected via a linear layer to match the decoder's channels.

Transformer decoder. The decoder comprises multiple Transformer decoder layers [35] that simultaneously process N embeddings of dimension d_{model}, $\mathcal{Q} \in \mathbb{R}^{N\times d_{\text{model}}}$, through multi-head self-attention and cross-attention mechanisms. Following established practices in [32, 40], these input embeddings are learnable positional embeddings which we refer to as *site queries*. The attention mechanisms enable the decoder to perform global reasoning over all potential binding sites while incorporating contextual information from the residue-level protein features output by the encoder.

Segmentation module. We process the N *site queries* through a linear classifier followed by the softmax activation to generate class probabilities $\{p_i = (p_i^{site}, p_i^{\emptyset}) | p_i^{site}, p_i^{\emptyset} \in [0, 1]\}_{i=1}^{N}$. Here, the classifier predicts an additional $\emptyset$ (non-binding) category to indicate when a *query* does not correspond to any actual binding site. For mask prediction, the *site queries* $\mathcal{Q} \in \mathbb{R}^{N\times d_{\text{model}}}$ are converted to N mask embeddings $\mathcal{E}_{\text{mask}} \in \mathbb{R}^{N\times d_{\text{model}}}$ by a MLP. Finally, the binary mask prediction for each *query* is computed via dot-production between the i-th mask embedding and the residue-level protein features $\mathcal{F} \in \mathbb{R}^{L\times d_{\text{model}}}$, followed by a sigmoid activation:

$$m_i[j] = \text{sigmoid}\left(\mathcal{E}_{\text{mask}}[i, :] \cdot \mathcal{F}[j, :]^T\right) \tag{5}$$

4 Rethinking the Evaluation Metrics for Binding Site Detection

DCC (Distance between the predicted binding site center and the true binding site center) and **DCA** (Shortest distance between the predicted binding site center and any heavy atom of the ligand) are the two most widely-used metrics for binding site detection. A binding site prediction is considered *successful* when its DCC or DCA value is below a predetermined threshold. Previous works [11, 37, 41, 14] quantify prediction performance via the **Success Rate**, defined as the ratio of *successful predictions* to the total number of ground truth sites:

$$\text{Success Rate (DCC or DCA)} = \frac{|\{\text{Predicted sites} \mid \text{DCC or DCA<threshold}\}|}{|\{\text{Ground truth sites}\}|} \tag{6}$$

However, these metrics suffer from two critical limitations: **(Limitation 1)** They disregard the prediction scores or ranks, and predictions may be double-counted due to the absence of proper matching criteria (Figure 4 A and Table S1). **(Limitation 2)** They only evaluate the center of

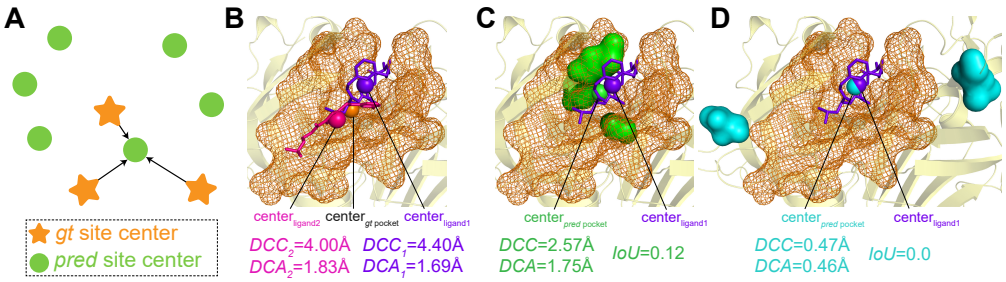

Figure 4: **DCC or DCA failure cases. (A)** Repeated counting of the same predicted site since absence of matching. **(B)** Different ligands bound to the same site lead to deviations in DCC or DCA calculations. **(C-D)** Failed predictions classified as successful by DCC or DCA but below the IoU threshold.

binding sites and are ligand-dependent (DCC typically considers the ligand center as site center). Since different ligands can bound to one binding site, relying solely on ligand-centered evaluation causes these metrics to completely miss key structural properties such as the shape, size, and residue composition of the binding site. It leads to evaluation failures in certain scenarios (Figure 4 B-D), disregard the crucial information required for downstream tasks (Appendix A).

The quantitative analysis of the DCC and DCA metrics is provided in Appendix G to further substantiate their evaluative flaws. The analysis reveals that approximately 20% of proteins are subject to double-counting during evaluation. Furthermore, the measured mean ground truth DCC (2.15 Å, 92.65% < 4 Å) and DCA (1.57 Å, 98.88% < 4 Å) exhibit a significant deviation from the ideal value of 0, revealing a systematic bias inherent to these metrics. These results directly correspond to the previously discussed limitations and confirm that DCC and DCA metrics significantly distort model performance assessment.

Previous works have recognized these limitations. To address **Limitation 1**, a common approach is to calculate the DCC or DCA for either the **top-n** or **top-(n+2)** predicted binding sites [11, 13, 14], where n is the number of ground truth sites. For **Limitation 2**, DeepSurf [41] and Utgés *et al.* [42] propose computing the **IoU** between the predicted and ground truth binding site residues. Given two binding sites $m_A, m_B \in \{0,1\}^L$, where L is the protein sequence length, the IoU of two sites is defined as:

$$\text{IoU}(m_A, m_B) = \frac{\text{sum}(m_A \& m_B)}{\text{sum}(m_A | m_B)}, \text{where } m_A, m_B \in \{0,1\}^L \tag{7}$$

However, these methods fail to address the core issues, as they still lack proper matching between predicted and ground truth sites, and the **top-n** or **top-(n+2)** metrics introduce information leakage.

To overcome these limitations, we propose to calculate the **Average Precision (AP)** metric based on the **residue-level IoU** as a fair metric for method evaluation. We calculate AP as follows: First, we sort all predictions by confidence scores. Then, we match each ground truth site to the predicted site with the highest score and residue-level IoU above a predetermined threshold, enforcing a one-to-one assignment constraint. Finally, we compute AP as the area under the interpolated precision-recall curve following COCO evaluation protocols [43], which is widely used in object detection. The pseudo-code for AP calculation is provided in Appendix K. The AP metric offers two significant advantages: (1) the residue-level IoU enables accurate shape and size comparison between binding sites; (2) the one-to-one matching scheme inherently prevents double-counting of predictions.

5 Experiments

5.1 Settings

Dataset. UniSite-DS is used for training and validation. We employ MMSeq2 [44] to ensure that no test UniProt sequence has similarity above 0.9 to any sequence in the training set. For each UniProt sequence, we select the PDB structure with the highest sequence identity as the representative structure. Additionally, we compare UniSite with baseline methods on widely-used binding site benchmark datasets, HOLO4K [11] and COACH420 [11]. Following DeepSurf [41] and EquiPocket [13], we use the *mlig* subsets of HOLO4K and COACH420 for evaluation. Since

our models are trained under a UniProt-centric schema, we only consider single-chain structures, denoting the test datasets as HOLO4K-sc and COACH420 (all structures in COACH420 are originally single-chain). All test UniProt entries are strictly excluded from the training set. More details are provided in Appendix F.

Implementation details. We set $d_{\text{model}} = 256$ by default. The transformer encoder consists of 6 standard Transformer encoder layers with a feed-forward dimension 1024 and the dropout rate of 0.1. We employ 6 Transformer decoder layers following the architecture of DETR [32]. By default, We use 32 *site queries*, where each query is associated with a learnable positional encoding and a zero-initialized query embedding. The multi-layer perceptron in the segmentation module consists of 2 hidden layers with 256 channels. For mask prediction, we use a combination of BCE loss and dice loss [34]:

$$\mathcal{L}_{\text{mask}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} \tag{8}$$

where $\lambda_{\text{bce}} = \lambda_{\text{dice}} = 5.0$. The classification loss weight λ_{cls} is set to 2.0, and we downweight the classification loss by a factor of 10 when $c_i^{gt} = \emptyset$ to mitigate class imbalance. Following DETR [32], we apply segmentation modules which share the same weights after each decoder layer, and supervise their predictions by the set prediction loss. We optimize the model using the AdamW optimizer [45] with a learning rate of 1.0×10^{-4} and a weight decay factor of 0.05. For the structural encoder, we implement the GearNet-Edge network following the origin paper [38] without specialized feature engineering. Notably, we train the GearNet-Edge network from scratch rather than loading pre-trained weights. All models are trained on 8 NVIDIA RTX 4090 GPUs.

Our method predicts the associated residues along with a confidence score for each binding site. For applications requiring binding site centers, we compute these as the centroids of the convex hull encompassing all atoms within each predicted binding site [14, 46].

Table 1: **Results on UniSite-DS.** We highlight the top two performing methods for each metric in bold. [a] Fpocket-rescore denotes sites initially predicted by Fpocket and subsequently rescored by P2Rank. [b] VN-EGNN only outputs centers of predicted sites. For each center, We include the residues within a 9Å radius, which has the best AP performance (Appendix H).

Method	Type	Input	$AP_{0.3}\uparrow$	$AP_{0.5}\uparrow$
Fpocket [8]	Geometry-based	structure	0.1836	0.1017
Fpocket-rescore[a]	Machine-learning	structure + Fpocket result	0.5075	0.2349
P2Rank [11]		structure	0.5056	0.2157
DeepPocket [12]	CNN-based	structure + Fpocket result	0.4273	0.2334
GrASP [14]	GNN-based	structure	0.4469	0.2848
VN-EGNN[b] [15]		structure	0.1621	0.0705
UniSite-1D	Ours	**sequence**	**0.5121**	**0.3033**
UniSite-3D		structure	**0.5603**	**0.3835**

Evaluation metrics. For comprehensive evaluation, we compare IoU-based AP and traditional DCC or DCA metrics in HOLO4K-sc and COACH420. Since a Uniprot-centric data entry can contain ligands from multiply PDB structures, coordinate-dependent metrics like DCC and DCA become unsuitable due to potential inconsistencies in ligand spatial arrangements. Consequently, we merely use IoU-based AP metric on UniSite-DS. Following EquiPocket [13], we set the DCC or DCA threshold to 4Å, and compute the DCC or DCA success rate of **top-n** predictions. We calculate AP using IoU thresholds of 0.3 and 0.5.

5.2 Results on UniSite-DS

The results on UniSite-DS are shown in Table 1. The geometry-based method Fpocket [8] exhibits inferior performance since it merely considers the geometry and electronegativity. P2Rank [11] achieves better results by extracting the protein surface features with Random Forest. DeepPocket [12] utilizes 3D-CNN to rescore and refine Fpocket predictions, improving the preformance in $AP_{0.5}$. Notably, Fpocket-rescore, which combines Fpocket's initial predictions with P2Rank's re-ranking, surpasses both P2Rank and DeepPocket, highlighting the importance of proper scoring in binding site detection as captured by our AP metric. For graph models, GrASP [14] achieves further improvements in $AP_{0.5}$ by employing graph attention networks. However, VN-EGNN [15] performs poorly under

AP metrics, because it only outputs predicted binding site centers, discarding structural properties (shape and size) or residue identification. For evaluation purposes, we include the residues within a 9Å radius of each predicted center, which has the best AP performance (Appendix H).

Trained on the UniProt-centric dataset, UniSite-1D outperforms all baseline methods without protein structure, demonstrating remarkable capability for structure-free binding site detection, particularly valuable for site-aware protein–ligand docking (Appendix A). UniSite-3D further improves the performance remarkably by incorporating structure information. The above observations not only validate the effectiveness of our methods, but also reveal the significant statistical biases inherent in previous PDB-centric datasets, which limit the performance of prior methods.

5.3 Results on HOLO4K-sc and COACH420

The results on HOLO4K-sc and COACH420 are shown in Table 2. Fpocket [8], P2Rank [11] and DeePocket [12] exhibit a consistent performance ranking across different datasets and metrics. The improvement achieved by Fpocket-rescore is also consistently evident. These indicate the concordance between our proposed IoU-based AP and traditional DCC or DCA metrics. Besides, the AP metric demonstrates superior discriminative power. On HOLO4K-sc, while DeepPocket and GrASP [14] show almost identical performance in $DCA_{top\text{-}n}$ (< 0.01), they diverge substantially (> 0.10) in $AP_{0.3}$. Similarly, on COACH420, performance differences among Fpocket-rescore, P2Rank, and GrASP are more pronounced under AP evaluation than with DCC metrics. As an exception, VN-EGNN [15] performs well in DCC or DCA while performing poorly under AP, as it merely predicts the binding site centers, discarding structural properties and reisidue identification. Notably, it is problematic since both the structural properties and the residue identification of binding sites are critical for downstream tasks (Appendix A). Since UniSite-DS is a UniProt-centric dataset while HOLO4K-sc and COACH420 are both PDB-centric, there is a training-test gap for UniSite-1D/3D. Even so, both UniSite-1D and UniSite-3D maintain strong performance on the two PDB-centric benchmarks across all evaluation metrics, demonstrating the effectiveness of our framework. More results are provided in Appendix J.

Table 2: **Results on HOLO4K-sc and COACH420.** We highlight the top two performing methods for each metric in bold. [a] Fpocket-rescore denotes sites initially predicted by Fpocket and subsequently rescored by P2Rank. [b] VN-EGNN only outputs centers of predicted sites. For each center, We include the residues within a 9Å radius, which has the best AP performance (Appendix H).

Method	HOLO4K-sc			COACH420		
	$AP_{0.3}\uparrow$	$DCC_{top\text{-}n}\uparrow$	$DCA_{top\text{-}n}\uparrow$	$AP_{0.3}\uparrow$	$DCC_{top\text{-}n}\uparrow$	$DCA_{top\text{-}n}\uparrow$
Fpocket [8]	0.2711	0.3076	0.4382	0.2106	0.2708	0.4107
Fpocket-rescore[a]	0.5899	0.5183	0.7654	0.5602	0.4405	0.7113
P2Rank [11]	0.6011	0.5300	**0.8188**	0.6188	0.4643	0.7411
DeepPocket [12]	0.5415	0.4925	0.7369	0.5184	0.3958	0.6756
GrASP [14]	0.6668	0.5131	0.7416	**0.7150**	**0.4851**	**0.7620**
VN-EGNN[b] [15]	0.2606	**0.5861**	0.6999	0.2637	**0.5446**	**0.7530**
UniSite-1D (ours)	**0.6867**	0.5538	0.7692	0.5921	0.4554	0.7351
UniSite-3D (ours)	**0.7091**	**0.5716**	**0.7879**	**0.7196**	0.4702	0.7381

Table 3: **Effect of sequence similarity.** The second column indicates the sequence identity between training sets and test sets.

Method	Similarity	$AP_{0.3}\uparrow$	$AP_{0.5}\uparrow$
UniSite-1D	<0.9	0.5121	0.3033
UniSite-3D		0.5603	0.3835
UniSite-1D	<0.7	0.5056	0.2945
UniSite-3D		0.5579	0.3734
UniSite-1D	<0.5	0.4338	0.2243
UniSite-3D		0.4677	0.2801

Table 4: **Effect of site queries.** This table shows results of UniSite-3D trained with a varying number of site queries.

# of queries	$AP_{0.3}\uparrow$	$AP_{0.5}\uparrow$
16	0.5515	0.3795
32	0.5603	0.3835
48	0.5562	0.3861
64	0.5615	0.3867

5.4 Ablation study

Sequence similarity. Both the structure and function of proteins diverge with the decreasing of sequence similarity. It is necessary to evaluate our protein ligand binding site detection method across varying similarity thresholds. We employ MMSeqs2 [44] to partition UnSite-DS with three similarity thresholds: 0.5, 0.7 and 0.9, ensuring that no test protein exceeds the corresponding similarity to any training protein. Compared to threshold 0.9, UnSite-1D/3D exhibit only a slight decrease under threshold 0.7 (Table 3), indicating that our methods possess generalization ability. As proteins with sequence similarity below 0.5 typically belong to evolutionarily distant families and exhibit markedly different structural folds, we observe significant AP performance degradation for UniSite-1D and UniSite-3D under threshold 0.5.

Number of site queries. As shown in Table 4, UniSite-3D exhibits stable performance across varying numbers of site queries. we select 32 queries as our default configuration, considering both the computation cost and the coverage of ground truth binding sites (99.5% proteins in UniSite-DS have sites less than 20).

6 Concluding Remarks and Future Perspectives

Key Contributions. In this paper, we introduce UniSite-DS, the first UniProt-centric dataset of protein ligand binding sites, which systematically integrates all ligand binding sites across multiple PDB structures for each unique protein. UniSite-DS corrects the statistical bias in previously available PDB-centric datasets and methods, while significantly broadening the available data. To amend the discontinuous workflows in existing binding site detection methods, we proposed UniSite-1D/3D, two end-to-end methods supervised by set prediction loss with bijective matching. In addition, we introduce IoU-based AP as a more accurate evaluation metric. Extensive experiments on UniSite-DS and several benchmark datasets demonstrate that our frameworks achieve superior performance, and the IoU-based AP metric can provide a more accurate reflection of binding site prediction quality.

Limitations and Future Work. The current version of UniSite-DS involves manual curation to remove unreasonable entries. A promising direction for future work is to develop automated methods for repairing and reintegrating excluded data to further enhance the dataset's coverage and quality. Additionally, our current model design aims to demonstrate the effectiveness of the end-to-end ligand binding site learning framework, without incorporating specialized feature engineering. Future investigations could explore the inclusion of specialized feature engineering to further improve model performance and generalization ability.

Acknowledgements

We thank the reviewers for their constructive feedback, and Hannes Stärk, Chenyu Wang, Zuobai Zhang, Zaixi Zhang, and Bozitao Zhong for helpful discussions. Liwei Wang is supported by National Science and Technology Major Project (2022ZD0114902) and National Science Foundation of China (NSFC92470123, NSFC62276005).

References

[1] Ryosuke Nakashima, Keisuke Sakurai, Seiji Yamasaki, Kunihiko Nishino, and Akihito Yamaguchi. Structures of the multidrug exporter acrb reveal a proximal multisite drug-binding pocket. *Nature*, 480(7378):565–569, 2011.

[2] Fabien Vincent, Arsenio Nueda, Jonathan Lee, Monica Schenone, Marco Prunotto, and Mark Mercola. Phenotypic drug discovery: recent successes, lessons learned and new directions. *Nature Reviews Drug Discovery*, 21(12):899–914, 2022.

[3] HC Stephen Chan, Yi Li, Thamani Dahoun, Horst Vogel, and Shuguang Yuan. New binding sites, new opportunities for gpcr drug discovery. *Trends in biochemical sciences*, 44(4):312–330, 2019.

[4] Stephen J Campbell, Nicola D Gold, Richard M Jackson, and David R Westhead. Ligand binding: functional site location, similarity and docking. *Current opinion in structural biology*, 13(3):389–395, 2003.

[5] Shuya Li, Tingzhong Tian, Ziting Zhang, Ziheng Zou, Dan Zhao, and Jianyang Zeng. Pocketanchor: Learning structure-based pocket representations for protein-ligand interaction prediction. *Cell Systems*, 14(8):692–705, 2023.

[6] Alexander S Powers, Helen H Yu, Patricia Suriana, Rohan V Koodli, Tianyu Lu, Joseph M Paggi, and Ron O Dror. Geometric deep learning for structure-based ligand design. *ACS Central Science*, 9(12):2257–2267, 2023.

[7] Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024.

[8] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, 10:1–11, 2009.

[9] Michal Brylinski and Jeffrey Skolnick. A threading-based method (findsite) for ligand-binding site prediction and functional annotation. *Proceedings of the National Academy of sciences*, 105(1):129–134, 2008.

[10] Chi-Ho Ngan, David R Hall, Brandon Zerbe, Laurie E Grove, Dima Kozakov, and Sandor Vajda. Ftsite: high accuracy detection of ligand binding sites on unbound protein structures. *Bioinformatics*, 28(2):286–287, 2012.

[11] Radoslav Krivák and David Hoksza. P2rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10:1–12, 2018.

[12] Rishal Aggarwal, Akash Gupta, Vineeth Chelur, CV Jawahar, and U Deva Priyakumar. Deeppocket: ligand binding site detection and segmentation using 3d convolutional neural networks. *Journal of Chemical Information and Modeling*, 62(21):5069–5079, 2021.

[13] Zhewei Wei, Ye Yuan, Chongxuan Li, Wenbing Huang, et al. Equipocket: an e (3)-equivariant geometric graph neural network for ligand binding site prediction. In *Forty-first International Conference on Machine Learning*, 2023.

[14] Zachary Smith, Michael Strobel, Bodhi P Vani, and Pratyush Tiwary. Graph attention site prediction (grasp): identifying druggable binding sites using graph neural networks with attention. *Journal of chemical information and modeling*, 64(7):2637–2644, 2024.

[15] Florian Sestak, Lisa Schneckenreiter, Johannes Brandstetter, Sepp Hochreiter, Andreas Mayr, and Günter Klambauer. Vn-egnn: E (3)-equivariant graph neural networks with virtual nodes enhance protein binding site identification. *arXiv preprint arXiv:2404.07194*, 2024.

[16] Jérémy Desaphy, Guillaume Bret, Didier Rognan, and Esther Kellenberger. sc-pdb: a 3d-database of ligandable binding sites—10 years on. *Nucleic acids research*, 43(D1):D399–D404, 2015.

[17] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, 47(12):2977–2980, 2004.

[18] Peter Schmidtke, Catherine Souaille, Frédéric Estienne, Nicolas Baurin, and Romano T Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of chemical information and modeling*, 50(12):2191–2200, 2010.

[19] Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

[20] Feng He, Cheng-Guo Wu, Yang Gao, Sabrina N Rahman, Magda Zaoralová, Makaía M Papasergi-Scott, Ting-Jia Gu, Michael J Robertson, Alpay B Seven, Lingjun Li, et al. Allosteric modulation and g-protein selectivity of the ca2+-sensing receptor. *Nature*, 626(8001):1141–1148, 2024.

[21] Edda SF Matthees and Carsten Hoffmann. The ca2+-sensing receptor and the pocketome: comparing nature's complexity with human intervention in receptor modulation. *Signal Transduction and Targeted Therapy*, 9(1):173, 2024.

[22] Johannes Morstein, Victoria Bowcut, Micah Fernando, Yue Yang, Lawrence Zhu, Meredith L Jenkins, John T Evans, Keelan Z Guiley, D Matthew Peacock, Sophie Krahnke, et al. Targeting ras-, rho-, and rab-family gtpases via a conserved cryptic pocket. *Cell*, 187(22):6379–6392, 2024.

[23] UniProt Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, 2019.

[24] Jifa Zhang, Yinglu Zhang, Jiaxing Wang, Yilin Xia, Jiaxian Zhang, and Lei Chen. Recent advances in alzheimer's disease: Mechanisms, clinical trials and new drug development strategies. *Signal transduction and targeted therapy*, 9(1):211, 2024.

[25] Timothy P Hughes, Michael J Mauro, Jorge E Cortes, Hironobu Minami, Delphine Rea, Daniel J DeAngelo, Massimo Breccia, Yeow-Tee Goh, Moshe Talpaz, Andreas Hochhaus, et al. Asciminib in chronic myeloid leukemia after abl kinase inhibitor failure. *New England Journal of Medicine*, 381(24):2315–2326, 2019.

[26] Lauren M Slosky, Yushi Bai, Krisztian Toth, Caroline Ray, Lauren K Rochelle, Alexandra Badea, Rahul Chandrasekhar, Vladimir M Pogorelov, Dennis M Abraham, Namratha Atluri, et al. β-arrestin-biased allosteric modulator of ntsr1 selectively attenuates addictive behaviors. *Cell*, 181(6):1364–1379, 2020.

[27] Jigang Fan, Yaqin Liu, Ren Kong, Duan Ni, Zhengtian Yu, Shaoyong Lu, and Jian Zhang. Harnessing reversed allosteric communication: a novel strategy for allosteric drug discovery. *Journal of Medicinal Chemistry*, 64(24):17728–17743, 2021.

[28] Christos P Feidakis, Radoslav Krivak, David Hoksza, and Marian Novotny. Ahoj: rapid, tailored search and retrieval of apo and holo protein structures for user-defined ligands. *Bioinformatics*, 38(24):5452–5453, 2022.

[29] Helen Berman, Kim Henrick, and Haruki Nakamura. Announcing the worldwide protein data bank. *Nature structural & molecular biology*, 10(12):980–980, 2003.

[30] Sameer Velankar, José M Dana, Julius Jacobsen, Glen Van Ginkel, Paul J Gane, Jie Luo, Thomas J Oldfield, Claire O'Donovan, Maria-Jesus Martin, and Gerard J Kleywegt. Sifts: structure integration with function, taxonomy and sequences resource. *Nucleic acids research*, 41(D1):D483–D489, 2012.

[31] Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. *arXiv preprint arXiv:2503.00710*, 2025.

[32] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[33] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.

[34] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[37] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S Rose, and Gianni De Fabritiis. Deepsite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

[38] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023.

[39] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer, 2018.

[40] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems*, 2021.

[41] Stelios K Mylonas, Apostolos Axenopoulos, and Petros Daras. Deepsurf: a surface-based deep learning approach for the prediction of ligand binding sites on proteins. *Bioinformatics*, 37(12):1681–1690, 2021.

[42] Javier S Utgés and Geoffrey J Barton. Comparative evaluation of methods for the prediction of protein–ligand binding sites. *Journal of Cheminformatics*, 16(1):126, 2024.

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*, 2014.

[44] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1):2542, 2018.

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[46] R Tyrrell Rockafellar. Conjugate convex functions in optimal control and the calculus of variations. *Journal of Mathematical Analysis and Applications*, 32(1):174–222, 1970.

[47] Marcel L Verdonk, Jason C Cole, Michael J Hartshorn, Christopher W Murray, and Richard D Taylor. Improved protein–ligand docking using gold. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003.

[48] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[49] Zhirui Liao, Ronghui You, Xiaodi Huang, Xiaojun Yao, Tao Huang, and Shanfeng Zhu. Deepdock: enhancing ligand-protein interaction prediction by a combination of ligand and structure information. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 311–317. IEEE, 2019.

[50] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023.

[51] Patrick Bryant, Atharva Kelkar, Andrea Guljas, Cecilia Clementi, and Frank Noé. Structure prediction of protein-ligand complexes from sequence information with umol. *Nature Communications*, 15(1):4536, 2024.

[52] Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

[53] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[54] Iambic Therapeutics. Transforming computational drug discovery with neuralplexer2. *https://www.iambic.ai/post/np2*, 2024. Accessed: 2025-05-01.

[55] Manfred Hendlich, Friedrich Rippmann, and Gerhard Barnickel. Ligsite: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, 15(6):359–363, 1997.

[56] Jie Liang, Clare Woodward, and Herbert Edelsbrunner. Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. *Protein science*, 7(9):1884–1897, 1998.

[57] Daniele Toti, Le Viet Hung, Valentina Tortosa, Valentina Brandi, and Fabio Polticelli. Librawa: a web application for ligand binding site detection and protein function recognition. *Bioinformatics*, 34(5):878–880, 2018.

[58] Radoslav Krivák and David Hoksza. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. *Journal of cheminformatics*, 7:1–13, 2015.

[59] Zhao Yang, Bing Su, Jiahao Chen, and Ji-Rong Wen. Interpretable enzyme function prediction via residue-level detection. *arXiv preprint arXiv:2501.05644*, 2025.

[60] Franck Da Silva, Jeremy Desaphy, and Didier Rognan. Ichem: a versatile toolkit for detecting, comparing, and predicting protein–ligand interactions. *ChemMedChem*, 13(6):507–510, 2018.

[61] Clemens Isert, Kenneth Atz, and Gisbert Schneider. Structure-based drug design with geometric deep learning. *Current Opinion in Structural Biology*, 79:102548, 2023.

A Impact of Binding Site Accuracy on Downstream Tasks

The accuracy of protein–ligand binding site detection is critical for downstream tasks. Taking molecular docking as an example (Figure S1), a comparison of different methods leads to the following conclusions: **(1) The definition of binding site residues can strongly impact the docking performance**, highlighting the importance of our proposed IoU-based AP metric. **(2)** Docking methods that do not specify binding sites (blind docking) **show significant performance improvements when provided with binding site information**, emphasizing the importance of accurate binding site identification.

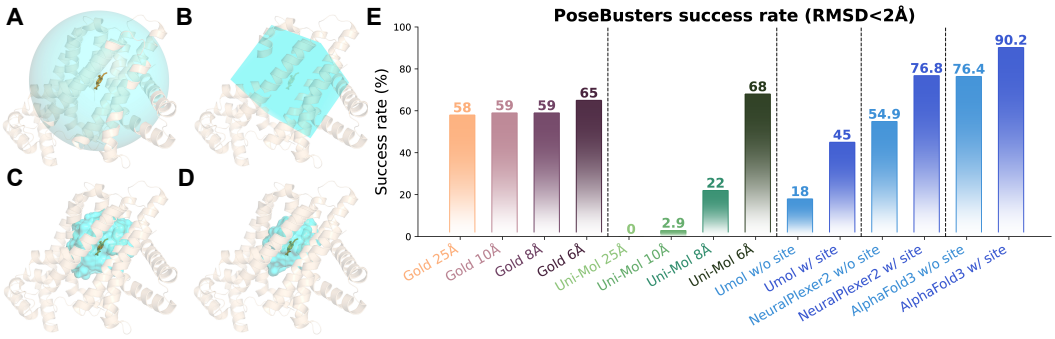

Figure S1: **The significant impact of binding site detection on molecular docking. (A)** Gold [47] defines the binding site using a sphere. **(B)** AutoDock Vina [48] defines the binding site using a cube. **(C)** DeepDock [49] and **(D)** Uni-Mol [50] identify the binding site by applying a fixed radius around the ligand. **(E)** Docking success rates on the PoseBusters dataset under different binding site configurations. Docking success rate is defined as the proportion of predictions with an RMSD less than 2Å. Data sourced from [51, 52, 53, 54].

B UniSite-DS Curation Workflow and Representative Manual Inspection Case Studies

As shown in Figure S2, the UniSite-DS workflow comprises three main components: (I) dataset curation, (II) quality control, and (III) manual inspection. Below, we describe two representative types of cases encountered during manual inspection that required special consideration:

1. **Case 1. Supramolecular assemblies across multiple protein subunits.** The Photosystem I-LHCI Supercomplex is a core component of the photosynthetic machinery, consisting of multiple subunits such as UniProt ID: P05310 (Photosystem I P700 chlorophyll a apoprotein A1, PDB: 7dkz_A), Q41038 (Chlorophyll a-b binding protein, 7dkz_B), and Q32904 (Chlorophyll a-b binding protein 3, 7dkz_C). Although each of these proteins binds a large number of Chlorophyll a molecules, they function as part of a tightly integrated supramolecular assembly. Therefore, they are not suitable to be included as independent entries in the UniSite-DS database and have been excluded in the current version.

2. **Case 2. Large composite cavities jointly formed by multiple ligands.** Trypanothione reductase (UniProt ID: Q389T8) is a key enzyme in *Trypanosoma brucei* that specifically catalyzes the reduction of trypanothione, functionally analogous to glutathione reductase in mammals. The ligands from structures PDB: 5s9x_A, 5s9t_A, and 2wov_C together form a large binding cavity. These ligand binding sites could potentially be merged. However, since each ligand actually occupies only a portion of the cavity rather than the entire cavity, whether such a composite cavity formed by different ligands should be considered a unified binding site often depends on system-specific definitions in the literature. As a result, entries that require further literature-based validation were excluded from the current version of the UniSite-DS database.

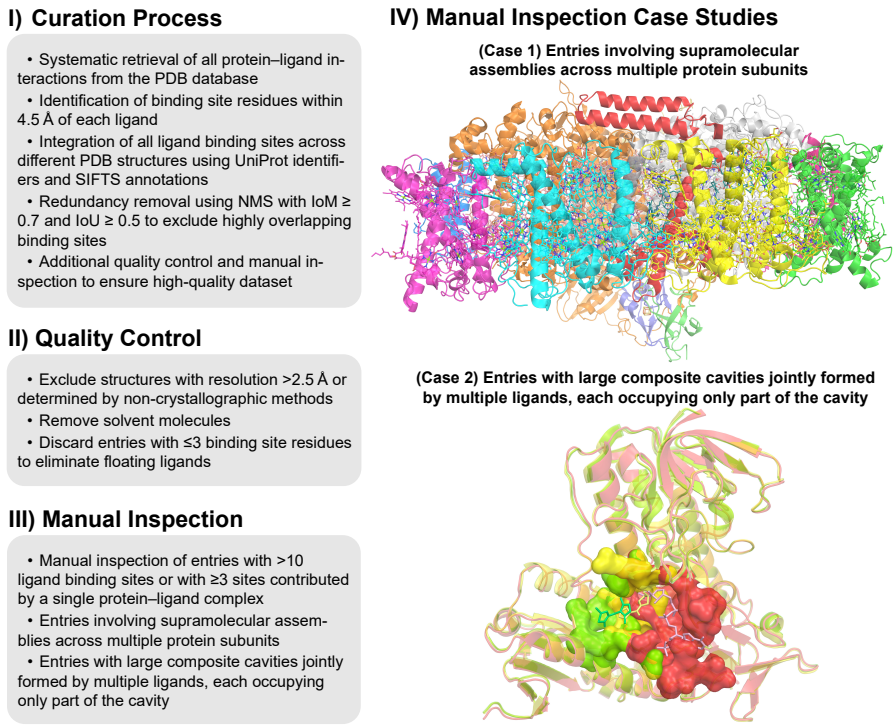

Figure S2: **Overview of the UniSite-DS workflow.** Workflow of (I) dataset curation, (II) quality control, and (III) manual inspection for UniSite-DS, together with (IV) representative manual inspection case studies.

C Related Work

Over the past several decades, numerous methods have been developed for detecting protein–ligand binding sites, accompanied by advances in techniques leveraging the geometric, physical and chemical features of proteins.

Early methods relied on traditional computational algorithms. Since most binding sites show up as cavities in protein 3D structures, geometry-based methods (Fpocket [8], LigSite [55]) identify and rank these hollow cavities through hand-crafted features like alpha spheres [56]. Template-based methods (FINDSITE [9] and LIBRA [57]) predict ligand binding sites by comparing the query protein with templates from known protein structure database. These methods typically generate a large number of predicted sites while performing poorly in ranking them.

Subsequent approaches like PRANK [58] and P2Rank [11] employ traditional machine learning methods, particularly Random Forest. Based on the predictions of Fpocket [8], PRANK assigns "ligandibility" scores, which denotes ligand binding potential, to candidate sites. P2Rank is a widely used method which integrates the geometric features of the protein surface with Random Forest Algorithm.

In recent years, deep learning methods have emerged for protein–ligand binding site detection. CNN-based approaches [12, 37, 41] treat protein structures as 3D images, applying 3D convolutional neural networks similar to those used in computer vision. Alternatively, GNN-based methods [15, 14, 13] utilize graph neural networks by constructing graphs incorporating both geometric and chemical features of proteins. Despite their improved performance, these methods typically adopt discontinuous workflows: they first perform semantic segmentation to generate binary masks of potential binding residues/atoms, then cluster these masks into discrete binding sites. This fragmented pipeline heavily depends on post-processing (e.g., clustering algorithms), inherently limiting end-to-end optimization and struggling with overlapping binding sites. DETR-based architectures have also been adapted for other protein tasks, such as ProtDETR [59], which performs enzyme function classification rather than binding site detection.

D Baseline Justification

We selected representative baseline methods that span the major methodological paradigms in ligand binding site detection. Fpocket [8] represents traditional geometry-based computational algorithms. P2Rank [11] is the most widely used machine learning approach. Among deep learning approaches, DeepPocket [12], GrASP [14], and VN-EGNN [15] were included as they represent different neural network architectures: DeepPocket employs 3D CNNs, while GrASP and VN-EGNN utilize GNNs.

For a fair and consistent comparison, we extracted ligand binding site residues from each baseline's output files according to their respective standard formats:

- **Fpocket:** For each predicted pocket, Fpocket outputs a file named `pocket{index}_atm.pdb`, which contains the atomic coordinates of the predicted binding site in PDB format. These atoms were directly used to identify the corresponding binding residues.
- **P2Rank:** P2Rank generates a `{name}.pdb_prediction.csv` file for each input structure. The `residue_ids` column specifies the chain IDs and residue IDs of residues constituting each predicted binding site. These identifiers were parsed to obtain the binding site residues.
- **DeepPocket:** DeepPocket refines and re-scores the pockets predicted by Fpocket, while maintaining the same output format.
- **GrASP:** GrASP outputs a `{name}_probs.pdb` file, where the `bfactor` column encodes the predicted binding probabilities of heavy atoms. Following the original publication, we filtered atoms based on their predicted scores and clustered them spatially into distinct binding sites.
- **VN-EGNN:** VN-EGNN produces a `prediction.csv` file containing four columns (`x`, `y`, `z`, `rank`) that specify the 3D coordinates and ranking of predicted pocket centers. As the method does not directly output residue-level predictions, binding site residues were obtained by including all residues within a defined radius, as described in Appendix H.

E Case Study of Classical Methods

As discussed in Section 1, 2, previous approaches are developed based on PDB-centric datasets, which introduce substantial statistical biases. We conduct a case study of these methods, and the results (Figure S3) indicate that, **for multi-site proteins, existing approaches are weak in distinguishing between different ligand binding sites**. This limitation motivated us to investigate the problems in existing approaches and to develop a new methodology.

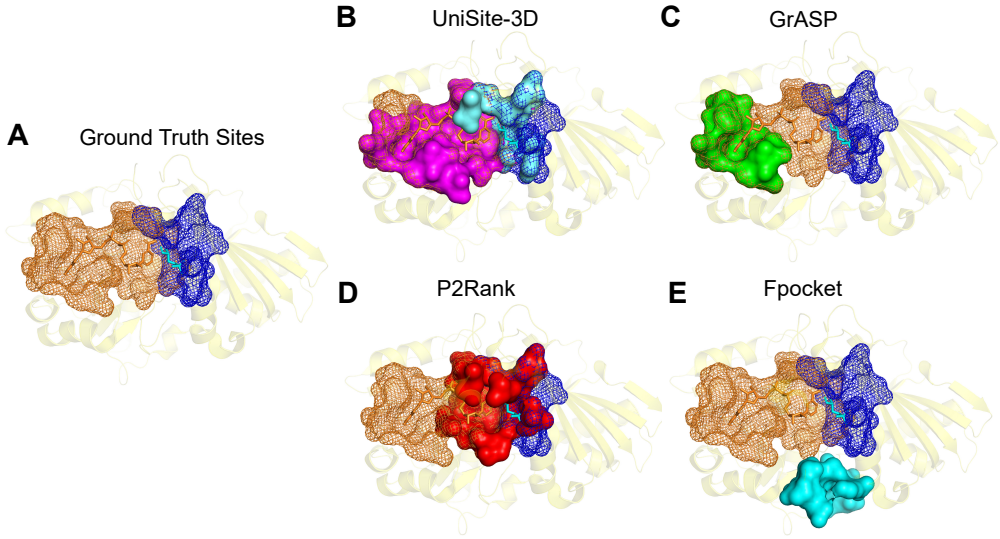

Figure S3: **Case study of classical methods.** For UniProt ID Q7YYQ9, the two ground truth binding sites are represented by dark blue and orange meshes, respectively. All predicted binding sites are shown as surfaces. **(A)** The two ground truth binding sites are colored in dark blue and orange. **(B)** The two binding sites predicted by our UniSite-3D method are colored in cyan and purple. **(C)** The single binding site predicted by the GNN-based method GrASP [14] is colored in green. **(D)** The single binding site predicted by the classical machine learning method P2Rank [11] is colored in red. **(E)** The single binding site predicted by the geometry-based method Fpocket [8] is colored in cyan.

F External Datasets

scPDB [16] is a famous dataset for protein–ligand binding site detection, commonly employed for training and validation in recent studies ([15, 14, 13]). scPDB provides both protein and ligand structures, accompanied by the structures of binding site extracted via VolSite [60]. Notably, only one binding site and one corresponding ligand are annotated for each data entry. In this work, we use the 2017 release of scPDB, which contains 17,594 structures and 5,550 unique proteins. (Source: http://bioinfo-pharma.u-strasbg.fr/scPDB/)

PDBBind [17] is a widely used dataset to study protein–ligand interaction, especially for protein–ligand docking [51, 61]. Similar to scPDB, PDBBind annotates one ligand structure and one binding site structure in each data entry. In this paper, we use the general set of v2020, the latest academic-free edition, which comprises 19,443 structures and 3,888 unique proteins. (Source: http://www.pdbbind.org.cn/download/)

HOLO4K and **COACH420** are two benchmark datasets utilized for protein–ligand binding site detection. Follow VN-EGNN [15], EquiPocket [13] and GrASP [14], we employ the *mlig* subsets of these two dataset, which contain explicitly specified relevant ligands. HOLO4K-*mlig* comprises 3,204 structures and 1,259 unique proteins, while COACH420-*mlig* covers 284 structures and 265 unique proteins. (Source: https://github.com/rdk/p2rank-datasets)

G Quantitative Analysis of Evaluation Flaws in DCC and DCA Metrics

In Section 4, we discussed two critical limitations in the design of the DCC and DCA metrics. Here, we conduct a quantitative analysis on the HOLO4K-sc benchmark to demonstrate the flawed evaluation introduced by these metrics.

Limitation 1. The absence of proper matching criteria may lead to double-counting of predictions (Figure 4 A). We quantified the proportion of proteins affected by double-counting during evaluation on the HOLO4K-sc benchmark (Table S1). The results reveal that DCC and DCA metrics suffer from widespread double counting artifacts, which significantly distort model performance assessment.

Table S1: **Double counting (DC) rate of DCC or DCA metrics on HOLO4K-sc.** We highlight the top two performing methods for each metric in bold. [a] Fpocket sites rescored by P2Rank. [b] Residues within 9Å of each VN-EGNN predicted center, which has the best AP performance (Appendix H).

Method	HOLO4K-sc	
	DC of $DCC_{top\text{-}n}$	DC of $DCA_{top\text{-}n}$
Fpocket [8]	18.80%	18.31%
Fpocket-rescore[a]	13.23%	13.66%
P2Rank [11]	18.98%	18.31%
DeepPocket [12]	13.53%	14.21%
GrASP [14]	16.29%	14.88%
VN-EGNN[b] [15]	12.80%	11.70%
UniSite-1D (ours)	**8.88%**	**8.94%**
UniSite-3D (ours)	**9.86%**	**10.04%**

Limitation 2. DCC or DCA only evaluate the center of binding sites and are ligand-dependent, which leads to evaluation failures in certain scenarios (Figure 4 B-D). For HOLO4K-sc, we calculated these metrics of the centroid of ground truth binding residues for each protein. The results indicate that the mean ground truth DCC is 2.15 Å (92.65% < 4 Å), and the mean ground truth DCA is 1.57 Å (98.88% < 4 Å). However, in principle, both DCC and DCA should ideally be 0 when evaluated using ground truth binding residues, indicating these metrics inherently contain systematic bias.

To mitigate some of DCC's inherent limitations, we defined a corrected metric, **DCC-residue**, which uses the center of ground truth binding residues rather than the ligand center for calculation. This modification resolves failure cases caused by ligand diversity in traditional DCC evaluation. As shown in Table S2, the corrected DCC-residue metric exhibits improved consistency with IoU-based AP in ranking the performance of different methods.

Table S2: **IoU-based AP, DCC-residue and DCC results on HOLO4K-sc.** We highlight the top two performing methods for each metric in bold. [a] Fpocket sites rescored by P2Rank. [b] Residues within 9Å of each VN-EGNN predicted center, which has the best AP performance (Appendix H).

Method	HOLO4K-sc		
	$AP_{0.3}$ ↑	DCC-residue$_{\text{top-}n}$ ↑	$DCC_{\text{top-}n}$ ↑
Fpocket [8]	0.2711	0.2982	0.3076
Fpocket-rescore[a]	0.5899	0.5005	0.5183
P2Rank [11]	0.6011	0.4972	0.5300
DeepPocket [12]	0.5415	0.4902	0.4925
GrASP [14]	0.6668	0.5379	0.5131
VN-EGNN[b] [15]	0.2606	0.5997	**0.5861**
UniSite-1D (ours)	**0.6867**	**0.6400**	0.5538
UniSite-3D (ours)	**0.7091**	**0.6264**	**0.5716**

H The AP Evaluation of VN-EGNN

Since VN-EGNN [15] outputs only the centers of predicted binding sites, it is non-trivial to identify the binding residues. In order to evaluate the IoU-based AP, we include the residues with a fixed radius for each predicted center (Figure S4). As shown in Table S3 , the radius of 9Å has the best performance across all datasets.

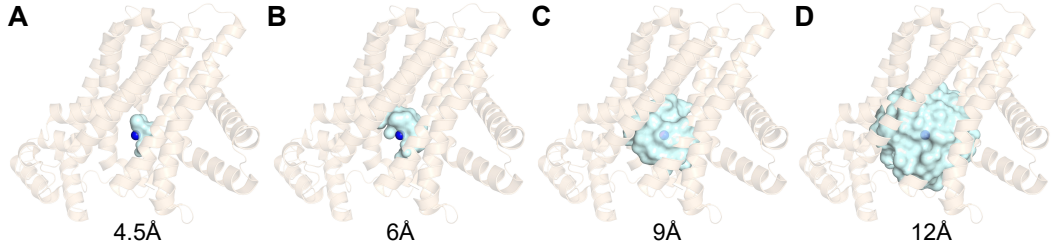

Figure S4: **Binding sites derived from the predicted center by VN-EGNN using different radii.** Binding site residues are visualized using surface representation: **(A)** residues within a 4.5Å radius; **(B)** residues within a 6Å radius; **(C)** residues within a 9Å radius; **(D)** residues within a 12Å radius.

Table S3: **AP evaluation results of VN-EGNN using different radii.**

Radius	UniSite-DS		HOLO4K-sc		COACH420	
	$AP_{0.3}$↑	$AP_{0.5}$↑	$AP_{0.3}$↑	$AP_{0.5}$↑	$AP_{0.3}$↑	$AP_{0.5}$↑
4.5Å	0.0054	0.0004	0.0049	0.0001	0.0072	0.0007
6Å	0.0894	0.0088	0.1290	0.0172	0.1814	0.0189
9Å	**0.1621**	**0.0705**	**0.2606**	**0.1346**	**0.2637**	**0.1138**
12Å	0.0087	0.0010	0.1411	0.0014	0.1444	0.0013

I The Recall Evaluation of Different Methods

The AP metric provides a comprehensive evaluation of binding site detection performance by considering both precision and recall. However, it is also important to evaluate how many true binding sites can be recovered under different IoU thresholds, since the goal of binding site detection is to identify novel and biologically meaningful sites/pockets. We computed the Recall of different methods across multiple IoU thresholds on the HOLO4K-sc dataset, and the results are summarized in Table S4. UniSite-3D achieves state-of-the-art Recall across all IoU thresholds.

Notably, the high recall obtained by Fpocket is due to its tendency to output nearly all potential cavities in a protein, rather than accurately identifying biologically relevant pockets. In fact, Fpocket tends to assign lower scores to true binding sites, making it challenging for biologists to distinguish meaningful pockets. This limitation has motivated many studies to develop rescoring strategies for Fpocket. While Recall reflects the potential of a method to detect biologically significant pockets, it does not account for the confidence score of each prediction. This limitation is addressed by the AP metric, which is why we adopt it as a fair and balanced criterion for evaluating different methods.

Table S4: **The AP and Recall results on HOLO4K-sc.** We highlight the top two performing methods for each metric in bold. [a] Fpocket sites rescored by P2Rank. [b] Residues within 9Å of each VN-EGNN predicted center, which has the best AP performance (Appendix H).

Method	HOLO4K-sc					
	$AP_{0.3}$	$AP_{0.5}$	$Recall_{0.3}$	$Recall_{0.5}$	$Recall_{0.7}$	$Recall_{0.9}$
Fpocket [8]	0.2711	0.1488	**0.8361**	0.5922	0.2130	0.0253
Fpocket-rescore[a]	0.5899	0.2847	0.8361	0.5922	0.2130	0.0253
P2Rank [11]	0.6011	0.2625	0.7814	0.5337	0.1868	0.0089
DeepPocket [12]	0.5415	0.2891	0.7514	0.5824	0.2584	0.022
GrASP [14]	0.6668	0.4126	0.7186	0.5374	0.2537	0.0159
VN-EGNN[b] [15]	0.2606	0.1346	0.7289	0.4874	0.0566	0
UniSite-1D (ours)	**0.6867**	**0.4595**	0.8212	**0.6199**	**0.3535**	**0.0824**
UniSite-3D (ours)	**0.7091**	**0.5446**	**0.8469**	**0.6901**	**0.4106**	**0.1039**

J Full Results on HOLO4K-sc and COACH420

Table S5: **Full results on HOLO4K-sc.** We highlight the top two performing methods for each metric in bold. [a] Fpocket sites rescored by P2Rank. [b] Residues within 9Å of each VN-EGNN predicted center, which has the best AP performance (Appendix H).

Method	HOLO4K-sc					
	$AP_{0.3}$	$AP_{0.5}$	$DCC_{\text{top-}n}$	$DCC_{\text{top-}n+2}$	$DCA_{\text{top-}n}$	$DCA_{\text{top-}n+2}$
Fpocket [8]	0.2711	0.1488	0.3076	0.4181	0.4382	0.5941
Fpocket-rescore[a]	0.5899	0.2847	0.5183	0.5941	0.7654	**0.8577**
P2Rank [11]	0.6011	0.2625	0.5300	0.5623	**0.8188**	**0.8652**
DeepPocket [12]	0.5415	0.2891	0.4925	0.5478	0.7369	0.7851
GrASP [14]	0.6668	0.4126	0.5131	0.5267	0.7416	0.7612
VN-EGNN[b] [15]	0.2606	0.1346	**0.5861**	0.6339	0.6999	0.7500
UniSite-1D (ours)	**0.6867**	**0.4595**	0.5538	**0.6400**	0.7692	0.8305
UniSite-3D (ours)	**0.7091**	**0.5446**	**0.5716**	**0.6470**	**0.7879**	0.8422

Table S6: **Full results on COACH420.** We highlight the top two performing methods for each metric in bold. [a] Fpocket sites rescored by P2Rank. [b] Residues within 9Å of each VN-EGNN predicted center, which has the best AP performance (Appendix H).

Method	COACH420					
	$AP_{0.3}$	$AP_{0.5}$	$DCC_{\text{top-}n}$	$DCC_{\text{top-}n+2}$	$DCA_{\text{top-}n}$	$DCA_{\text{top-}n+2}$
Fpocket [8]	0.2106	0.1219	0.2708	0.3750	0.4107	0.5714
Fpocket-rescore[a]	0.5602	0.2905	0.4405	0.5179	0.7113	**0.8333**
P2Rank [11]	0.6188	0.2618	0.4643	0.5000	0.7411	0.8034
DeepPocket [12]	0.5184	0.2512	0.3958	0.4821	0.6756	0.7560
GrASP [14]	**0.7150**	**0.4914**	**0.4851**	0.4970	**0.7620**	0.7917
VN-EGNN[b] [15]	0.2637	0.1138	**0.5446**	**0.6071**	**0.7530**	0.7768
UniSite-1D (ours)	0.5921	0.2998	0.4554	0.5238	0.7351	0.8006
UniSite-3D (ours)	**0.7196**	**0.3977**	0.4702	**0.5387**	0.7381	**0.8095**

K Pseudo-code for Average Precision Calculation

Algorithm 1 Average Precision Calculation

Input:
`prediction_list`: A list, where each element represents the predictions for one protein. Each element is of the form $\{(s_i, m_i)|s_i \in \mathbb{R}, m_i \in \{0,1\}^L\}_{i=1}^N$, where m_i represents the i-th predicted binding site as a binary mask of length L, and s_i denotes the i-th confidence score.
`ground_truth_list`: A list, where each element represents the ground truth (gt) binding sites for one protein. Each element is of the form $\{m_i^{gt}|m_i^{gt} \in \{0,1\}^L\}_{i=1}^{N_{gt}}$, where m_i^{gt} represents the i-th ground truth binding site as a binary mask of length L.
`iou_threshold`: IoU threshold for considering a prediction as True Positive.
Output: `AP`: Average Precision under the given IoU threshold.

Step 1: Determine TP (True Positive) or FP (False Positive) for each prediction

for `predictions_per_protein`, `ground_truths_per_protein` **in** ZIP(`prediction_list`, `ground_truth_list`) **do**
  Set all ground truths in `ground_truths_per_protein` as **unused**
  Sort `predictions_per_protein`=$\{(s_i, m_i)|s_i \in \mathbb{R}, m_i \in \{0,1\}^L\}_{i=1}^N$ by decreasing confidence score s_i
  for $i = 1$ to N **do**
    Find `ground_truth_j` that has the max residue-level IoU(m_j^{gt}, m_i) with `prediction_i`
    if IoU(m_j^{gt}, m_i) $>$ `iou_threshold` **and** `ground_truth_j` is **unused then**
      Mark `prediction_i` as TP
      Set `ground_truth_j` as **used**
    else
      Mark `prediction_i` as FP
    end if
  end for
end for

Step 2: Sort all predictions across all proteins by decreasing confidence scores

`all_predictions` $\leftarrow$ FLATTEN_ALL(`prediction_list`)
`all_ground_truths` $\leftarrow$ FLATTEN_ALL(`ground_truth_list`)
Sort `all_predictions` by decreasing confidence scores

Step 3: Calculate the precision-recall curve

Set `cum_TP`, `cum_FP` as 0, `precision_list` and `recall_list` as empty
for `prediction_i` **in** `all_predictions` **do**
  if `prediction_i` is marked as TP **then**
    `cum_TP` $\leftarrow$ `cum_TP` + 1
  else
    `cum_FP` $\leftarrow$ `cum_FP` + 1
  end if
  `precision_list`.append(`cum_TP` / (`cum_TP` + `cum_FP`))
  `recall_list`.append(`cum_TP` / LEN(`all_ground_truths`))
end for

Step 4: Calculate average precision

Calculate `AP` as the area under the precision-recall curve
return `AP`

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims reflect our contribution about dataset, methodology and evaluation metrics, which are illustrated in Section 2, Section 3 and Section 4 respectively.

   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: We discussed the limitations in Section 6.

   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: This paper does not include theoretical results.

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: We described the details of our model architecture, evaluation metrics, and experiment settings in Section 3-5.

   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: The dataset and codes will be made publicly available at `https://github.com/quanlin-wu/unisite`.

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: We described the details of our experiment settings and model implementation in Section 5.1.

   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 bars are not reported because it would be too computationally expensive.

   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: As described in Section 5.1, all experiments are conducted using 8 NVIDIA RTX 4090 GPUs (24G memory) for about 2 days.

   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: We have conducted with the NeurIPS Code of Ethics.

   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: Our work will propagate the development of structure based drug discovery.

    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: This paper poses no such risks.

    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: All assets used in this paper have been properly credited to their original creators or owners. Unless otherwise specified, all methods implemented in this work adhere to the MIT License. All datasets include source information or URLs, and their usage follows the licensing terms provided by the respective sources.

    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: [Yes]

    Justification: The dataset and codes will be made publicly available at `https://github.com/quanlin-wu/unisite`.

    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: This paper does not involve crowdsourcing nor research with human subjects.

    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: This paper does not involve crowdsourcing nor research with human subjects.

    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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.