GLPocket: A Multi-Scale Representation Learning Approach for Protein Binding Site Prediction

Peiying Li[1] , **Yongchang Liu**[1] , **Shikui Tu**[1,*] and **Lei Xu**[1,2,*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]Guangdong Institute of Intelligence Science and Technology, Zhuhai, Guangdong, China
{lipeiying, liuyongchang, tushikui, leixu}@sjtu.edu.cn
* corresponding author

Abstract

Protein binding site prediction is an important prerequisite for the discovery of new drugs. Usually, natural 3D U-Net is adopted as the standard site prediction framework to do per-voxel binary mask classification. However, this scheme only performs feature extraction for single-scale samples, which may bring the loss of global or local information, resulting in incomplete, artifacted or even missed predictions. To tackle this issue, we propose a network called GLPocket, which is based on the Lmser (Least mean square error reconstruction) network and utilizes multi-scale representation to predict binding sites. Firstly, GLPocket uses Target Cropping Block (TCB) for targeted prediction. TCB selects the local interested feature from the global representations to perform concentrated prediction, and reduces the volume of feature maps to be calculated by 82% without adding additional parameters. It integrates global distribution information into local regions, making prediction more concentrated on decoding stage. Secondly, GLPocket establishes long-range relationship of patches within the local region with Transformer Block (TB), to enrich local context semantic information. Experiments show that GLPocket improves by $0.5\% - 4\%$ on DCA Top-n prediction compared with previous state-of-the-art methods on four datasets. Our code has been released in https://github.com/CMACH508/GLPocket.

1 Introduction

Binding sites on the surface of 3D proteins are usually in the form of deep grooves or tunnels to accommodate small molecule drugs or ligands. Binding sites are also regarded as binding pockets, or binding cavities . Given a binding site, small molecule drugs can be designed and bound to the protein, so as to change the characteristics and biological functions of the protein. Thus, accurate detection of binding sites is the premise of drug discovery and drug design [Anderson, 2003; Patani and LaVoie, 1996].

It is a very challenging problem to detect the binding sites. The binding sites on proteins are much smaller than proteins themselves. The protein surface are very bumpy and may lead to many false positive sites. Examples of protein-ligand pairs are given in Fig. 1. We quantitatively calculate the ratio of the largest diameter of ligand to the largest diameter of protein. From diameter ratios below each figure, we can see that the pocket where the ligand is located is much smaller than protein, since the length of some ligands is about $1/10$ of the protein length. It is very difficult to accurately predict such small pockets for a given large protein structure.

Many methods have been proposed to tackle protein binding site detection problem. Traditional methods can be divided into three categories, i.e., geometric-based methods, template-based methods, and energy-based methods. Geometric-based methods, including Fpocket [Le Guilloux *et al.*, 2009], Ligsite-series [Hendlich *et al.*, 1997; Huang and Schroeder, 2006] and CriticalFinder [Nguyen *et al.*, 2017], predict binding sites according to geometric characteristics of protein surface, and then sort the candidate sites according to their druggable ability. Template-based methods, such as FindSite [Brylinski and Skolnick, 2008], search for the most similar protein from a database, and then assign the binding site of the hit protein to the query protein. These methods require a large number of proteins in the database and known locations of the binding sites for each protein. Energy-based methods [Ravindranath and Sanner, 2016; Ngan *et al.*, 2012; Hernandez *et al.*, 2009] are to find ligands that require minimal interaction energy to bind to proteins. These methods require multiple matches and massive ligand templates.

In the past few years, as a tremendous amount of 3D protein structures become available, machine learning models have been developed to learn from the data to predict the site location with promising performance. Prank [Kriválk and Hoksza, 2015] is a typical machine learning approach. It firstly uses Fpocket and concavity [Chen *et al.*, 2011] tools to assign and label pocket points according to physical and chemical properties. Random Forest is adopted to calculate the binding score of each pocket. Pockets with high scores are regarded as predicted sites.

In recent years, Deep Learning methods have been proposed for binding site detection with promising performance. DeepSite [Jiménez *et al.*, 2017] is the first work to adopt a 4-layer Convolutional Neural Network (CNN) as feature extractor to predict the location of binding sites. Protein is firstly voxelized into multi-channel 3D grids, and then the 3D grids

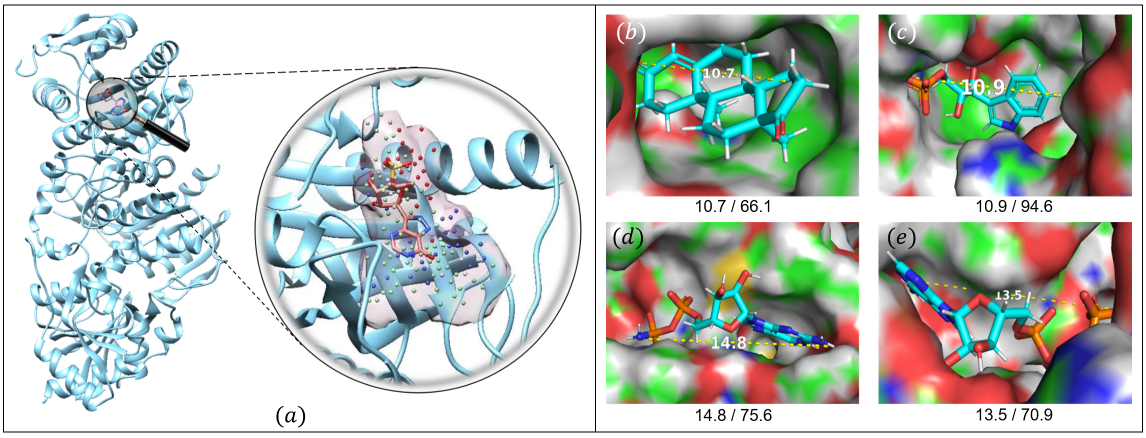

Figure 1: Figure (a) is a display of protein and ligand binding. The binding site is relatively very small, and the right image is an enlarged view of it. The stick object is the ligand, and the transparent body is the cavity, which is used to describe the space occupied by the ligand. The colorful dots are the voxel points inside the cavity. For texts in d_1/d_2 format below figure (b,c,d,e), d_1 represents the longest distance between ligand atoms, and d_2 represents the longest distance between protein atoms.

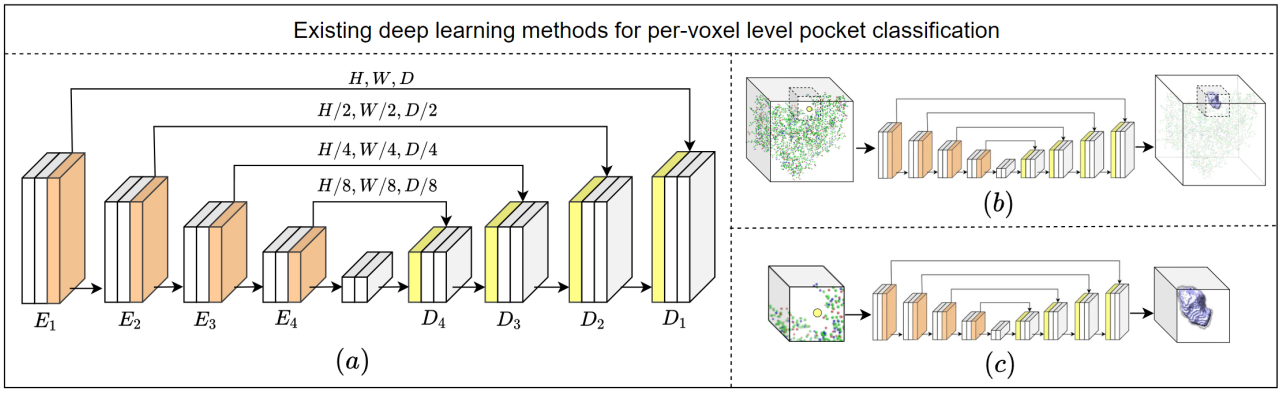

Figure 2: An overview of existing per-voxel mask classification method frameworks. (a) The natural U-Net architecture used for segmentation tasks. (b) Example of an end-to-end prediction architecture with global proteins as input, such as Kalasanty [Stepniewska-Dziubinska *et al.*, 2020]. (c) Example of framework for binary mask prediction with local regions of protein as input, such as DeepPocket [Aggarwal *et al.*, 2021] and RecurPocket ($\tau = 1$) [Li *et al.*, 2022].

are divided into multiple subgrids, which are used as the input of the CNN to obtain ligandability scores. DeepSite clusters the scores to screen out the likely ligandability regions, which are combined as the predicted site. DeepSurf [Mylonas *et al.*, 2021] takes local subgrids centered in each voxel point on the protein surface as network input to obtain ligandability scores. The scores are clustered to determine the final pockets. The classifications by DeepSite and DeepSurf are both made on local subgrids, not accurate enough to voxel level.

To do per-voxel classification, Kalasanty [Stepniewska-Dziubinska *et al.*, 2020] is the first method to employ 3D U-Net [Ronneberger *et al.*, 2015] as backbone, with the whole protein as input to predict a 3D binary mask. DeepPocket [Aggarwal *et al.*, 2021], RecurPocket [Li *et al.*, 2022] and RefinePocket [Liu *et al.*, 2023] take the local small-scale regions of a protein rather than the whole protein as input to predict pockets. They utilize Fpocket which predicts many candidate pockets with high recall but low precision, to generate candidate centers of each pocket. Then, they take subgrids centered in each candidate center as the input of 3D U-Net-based network for binary mask classification. The difference between RecurPocket ($\tau > 1$) and DeepPocket is that it builds a feedback link from decoder to encoder to predict pockets in a circular and progressive manner. Fig. 2 summarizes the network structures of the above mentioned methods.

Although the large-scale input can help Kalasanty to have a global view and do end-to-end binary mask predictions, it is difficult to focus on local regions of protein because the target is relatively very small, as shown in Fig. 1. As a result, it may lead to less discriminative features of small-scale regions and loss of details. On the contrary, DeepSite, DeepSurf, and DeepPocket take each small-scale subgrid of protein as input to do prediction, which brings the benefit that the networks are more inclined to notice small potential sites. However, this modeling doesn't consider the surrounding environment of the target and the global information of the protein structure. Thus the choice of the scale of the input to the model is a trade-off between the local attention and the global con-

sideration. The larger the scale, the greater amount of neighborhood information, but making it difficult to concentrate on small regions. The smaller the scale, the more attention will be paid to potential small sites, resulting in insufficient auxiliary features of the neighborhood morphology to determine the location and shape of pocket.

The Least mean square error reconstruction (Lmser) [Xu, 1993; Xu, 2019] was initially proposed by folding the autoencoder along the center hidden layer. It establishes forward skip connections and feedback connections between the paired layers of the encoder and decoder. Later, it evolved into a 2D or 3D CNN-based Lmser structure and has been proven to be better than U-Net in various tasks [Li *et al.*, 2019; Li *et al.*, 2022].

In this paper, we propose GLPocket based on Lmser architecture to capture the multi-scale representations of proteins to predict binding sites. In the encoding stage, we take the whole protein as network input to extract global representations. In order to make the predictions more concentrated and precise, a Target Cropping Block (TCB) is developed to extract local patterns of interest for targeted prediction. In the decoding stage, to fully capture the local context semantic information for refined prediction, Transformer Block (TB) is utilized to establish dependency relationships of patches within small-scale regions. Along with the global information transmitted from encoder, predictions for local regions are further refined.

Our contributions can be summarised as follow:

- A multi-scale Lmser network structure was proposed for binding site prediction on the surfaces of 3D protein structures. It learns global and local representations for targeted prediction.
- We devised a Target Cropping Block (TCB) to capture local features of interest for accurate details of the binding pockets. Moreover, TCB greatly reduces volume of the feature maps to be calculated of 82% without introducing additional parameters. We also built a Transformer Block (TB) to establish dependence relationships of patches within the local region, enriching the context semantic information.
- We demonstrated the effectiveness of GLPocket on four benchmark datasets. GLPocket outperforms related state-of-the-art methods in terms of various metrics.

2 Materials and Methods

2.1 Data Preparation

In our experiments, we adopt five publicly available datasets to evaluate our method. We use scPDB [Desaphy *et al.*, 2015] as training set, COACH420, HOLO4k, PDBbind [Wang *et al.*, 2005], SC6K as testing sets. We ensure that there is no intersection between the training set and the test sets. The details are summarized as follows.

ScPDB (v2017) is one of the largest protein-ligand pairs datasets. It is widely used as the training set for binding site prediction tasks [Jiménez *et al.*, 2017; Mylonas *et al.*, 2021; Stepniewska-Dziubinska *et al.*, 2020; Aggarwal *et al.*, 2021]. This dataset contains $16,612$ structures with $17,594$ binding sites. It also provides all-atom description of the protein-ligand pairs and their binding sites. We divide the dataset into ten parts and use one of them as validation dataset. PDBbind (v2020) contains $5,316$ protein-ligand complexes. After removing 18 proteins which is too large to be loaded, and $1,405$ complexes appeared in the training set, the remaining $3,893$ complexes were used for testing. COACH420 and HOLO4K have been used as testing sets to evaluate the performance of P2Rank [Radoslav and David, 2018]. We removed the proteins with invalid cavities, leaving 207 and $2,752$ structures in COACH420 and HOLO4K, respectively. SC6K dataset was added to the PDB (Protein Data Bank) from January 1st, 2018, until February 28th, 2020, for binding site detection. We used $2,378$ proteins screened by DeepPocket as testset.

2.2 The Architecture of GLPocket

An overview of the proposed GLPocket network is shown in the Fig. 3(a). The architecture applies Lmser as backbone, including Encoder Module, Decoder Module, Target Cropping Block (TCB) and Transformer Block (TB). Encoder Module contains one Pre block and four encoder blocks (E_i, $i \in [1,2,3,4]$). Decoder Module contains four decoder blocks (D_i, $i \in [1,2,3,4]$) and one $Post$ block. The details of TCB and TB are shown in the Fig. 3(b) and 3(c). Network structure details can be seen in the Fig. 3(d).

Encoder Module

Given an voxelized protein data $x \in \mathbb{R}^{C\times H\times W\times D}$, we firstly employ Pre block to extract preliminary characteristics of protein to prepare for subsequent encoder blocks. We utilize E_1 to further extract compact protein features, then apply TCB to choose interested region to focus on target proposal. Finally, E_2, E_3, E_4 are employed to extract proposal features to get g_4. The process is formulated as below:

$$\widetilde{f}_2 = TCB(E_1(Pre(x))), \tag{1}$$

$$g_4 = E_4(E_3(E_2(\widetilde{f}_2))), \tag{2}$$

where $\widetilde{f}_2$ is the local proposal feature maps obtained from global feature maps. As a result, the proposal representation integrates the global structural information and neighborhood information outside the region without expanding its size. Actually, it reduces the volume of feature maps to 18% of the original size.

Target Cropping Block (TCB)

Although the encoder is able to encode the pocket location in a rough manner from a global perspective, it is difficult for the decoder to predict the boundary details of a relatively small pocket on the protein surface. So we propose a Target Cropping Block (TCB) to capture small and interested local regions from the global feature maps as pocket proposals.TCB ensures more neighborhood information to enhance proposal representation without increasing its size.

The details of TCB are shown in Fig. 3(b). Given protein feature maps f_i ($i \in [1,2]$), TCB cuts a cube centered on the candidate center (in yellow) from f_i as a local proposal $\widetilde{f}_i$, and then passes $\widetilde{f}_i$ to subsequent block to extract features more intensively. In order to keep scale invariance, we set a

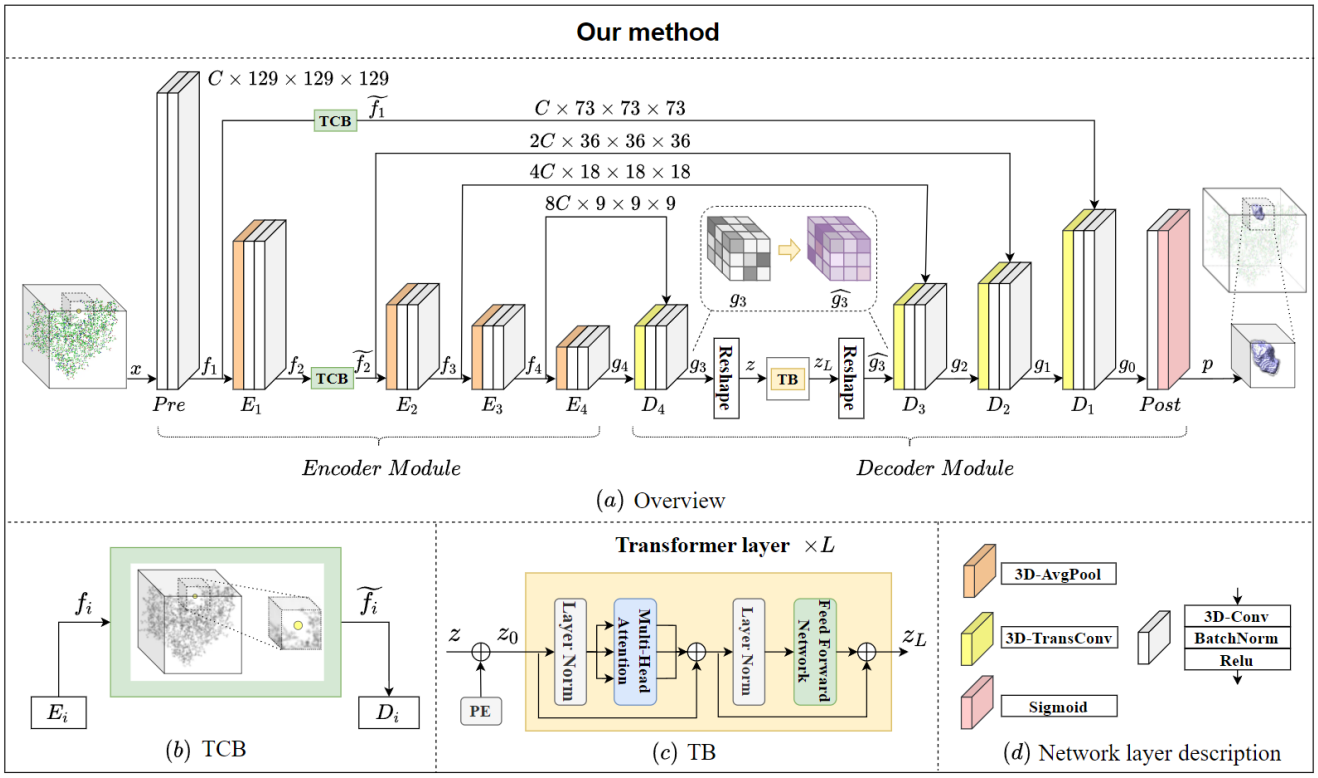

Figure 3: (a) An overview of proposed GLPocket. Encoder Module extracts large-scale samples, Target Cropping Block (TCB) is applied for obtaining small-scale features of interested proposal. Decoder Module intensively detects and reconstructs pocket in a small-scale region. Prediction is further refined by Transformer Block (TB). (b) Details of Target Cropping Block (TCB). (c) Details of Transformer Block (TB). (d) Network layer description.

cropping ratio for feature maps with different sizes. In this work, the cropping ratio of $\widetilde{f}_i$ to f_i is 0.56 for width, height and depth respectively. TCB does not introduce additional network parameters, and it also performs a 82% reduction of volume of feature maps by selecting regions of interest.

As shown in Fig. 3(a), the input of multi-channel 3D protein grid x is processed by Pre and E_1 blocks to get f_1 and f_2, later f_2 is processed by a TCB, and then $\widetilde{f}_2$ is passed to next encoder block. Like vanilla U-Net structure, encoder blocks need to pass multi-scale information to the corresponding decoder blocks for aggregating features of different semantic layers. Here, we transfer the proposal feature maps rather than the global protein feature maps, in order to help decoder make accurate local predictions. So f_1 and f_2 are processed by TCB and then passed to corresponding decoder blocks, while f_3 and f_4 are directly passed to corresponding decoder blocks.

Decoder Module

Decoder Module is responsible for predicting pocket of part of a protein rather than the whole protein, which is more purposely focused on small binding site. The input of Decoder Module includes not only forward propagation representation but also information transmitted from each encoder block. Following Eq. 3 & 4, D_i gradually predicts binding pocket by concatenating with the information passed from corresponding encoder blocks.

In addition, we utilize Transformer Block (TB) to model the long distance dependency relationship in proposal space after D_4 according to Eq. 3. Prediction result p can be get from $Post$ block following Eq. 4 and 5.

$$\widehat{g_3} = TB(D_4(g_4, f_4)), \tag{3}$$

$$g_0 = D_1(D_2(D_3(\widehat{g_3}, f_3), \widetilde{f}_2), \widetilde{f}_1), \tag{4}$$

$$p = Post(g_0), \tag{5}$$

where p is the predicted result and represents the probability that each voxel belongs to binding pocket.

Transformer Block (TB)

When processing the global feature maps of protein, Transformer Block (TB) will fall into a dilemma of high computational complexity due to long token sequence. To solve this problem, we put TB after Encoder Module to process feature representation with small size in the hidden space, so that the complexity can be reduced. But in high-level space, proposal representation is too compact and abstract to establish the long-distance dependencies of patches. Therefore, as shown in Fig. 3(a), after the first decoder block (D_4), where the resolution is increased, we apply TB for volumetric space and depth modeling.

The details of TB is shown in Fig. 3(a) and 3(c). Given the feature maps $g_3 \in \mathbb{R}^{K\times h\times w\times d}$, we flat and permute it into $z \in \mathbb{R}^{N\times K}$ as patch embeddings, where N is the number of tokens ($N = h \times w \times d$). Since spatial location relationship between different patches is the important information for comprehensive representation, we add learnable position embeddings (PE) to patch embeddings as follows:

$$z_0 = z + c_{pe}, \;\; c_{pe} \in \mathbb{R}^{N\times K} \tag{6}$$

where $z_0 \in \mathbb{R}^{N\times K}$ is the feature embeddings and is used as the input of TB, c_{pe} denotes position encoding.

The TB is composed of Transformer Layer repeated for L times. The yellow area in the Fig. 3(c) is Transformer Layer, which contains LayerNorm (LN), Multi-Head Attention (MHA), and Feed Forward Network (FFN). The result of each Transformer Layer z_l ($l \in [1, 2, 3, ..., L]$) can be get through the following equation,

$$\widehat{z_l} = z_{l-1} + MHA(LN(z_{l-1})), \tag{7}$$

$$z_l = \widehat{z_l} + FFN(LN(\widehat{z_l})), \tag{8}$$

where z_l is the output of l-th Transformer Layer. Finally, we reshape z_L to get volumetric data $\widehat{g_3}$.

2.3 Implementation Details

GLPocket is implemented in PyTorch and trained for 30 epochs with a batch size of 12 on 3 A100 GPUs. SGD optimizer was applied to train the model. The learning rate is set to 0.001 and remains the same. The binary cross entropy loss is employed to optimize our network.

2.4 Evaluation Metric

We use three metrics to evaluate the performance of models. The metrics include:

- **DCA** is the distance between the predicted pocket center and closest ligand atom. When distance is less than threshold, the prediction is considered to be correct, otherwise is wrong.
- **DCC** is the distance between the centers of predicted pocket and label. Prediction with DCC less than threshold is considered successful.
- **DVO** is the ratio of the overlap of predicted pocket and corresponding label to the union of their volumes.

In our experiments, we set threshold to 4Å as in related works. Both DCA success rate and DCC success rate are the ratio of the number of successfully predicted pockets to the total number of pockets. Under the DCC metric, the DVO is calculated for the correctly predicted pockets, otherwise, the DVO is set to 0 for the unpredicted pockets.

3 Results

3.1 Ablation Study

To analyze the efficiency of two blocks, Target Cropping Block (TCB) and Transformer Block (TB), we perform several ablation experiments. Models are evaluated under four test datasets.

Target Cropping Block

Target Cropping Block (TCB) is to select interested local feature from global feature representation for targeted prediction. Specifically, in encoding stage, according to the given candidate center and crop ratio, the protein feature maps are cropped for targeted predictions. This facilitates subsequent decoder blocks to extract local features more finely.

To further analyze TCB, we visualized the middle feature maps of GLPocket with and without TCB, as shown in Fig. 4. We can see that GLPocket without TCB introduces more artifacts in the output. GLPocket with TCB has a cleaner and more concentrated output with clearer boundaries. We can observe that the protein regions of with TCB have higher activate values in f_1 and f_2 than those without TCB, which leads to the features of two regions (empty region and protein region) to be more discriminative than those without TCB. This is mainly because GLPocket without TCB has limited local view. It is difficult to distinguish between proteins and empty areas. This problem is solved by TCB by selecting interested features from the global representation.

Furthermore, comparing f_3 of GLPocket with and without TCB, we can see that although they are the feature maps extracted from the same region of a protein, the prediction of with TCB is closer to the label than that without TCB. This is because the proposal integrates the global distribution information, making GLPocket with TCB more accurate to predict the shape of the pocket.

We also calculate the DCC and DVO of GLPocket with and without TCB, as shown in Tab. 1. Comparing the first two rows, TCB further improves prediction performance of model without increasing additional parameters.

Transformer Block

We also conduct comparative experiments to verify the effectiveness of Transformer Block (TB). TB aims to capture the spatial dependency within proposal to enrich local context semantic information.

The quantitative results are presented in Tab. 1. Comparing the first and third rows, GLPocket with TB, with only a few network parameters added, shows great superiority in both DCC and DVO metrics with significant improvements. This is mainly because the dependency between patches within proposal can be captured early in the decoding phase by using attention mechanism of TB. The subsequent decoder blocks can perform step-by-step targeted refinement on the local proposal based on the enhanced features. The result clearly reveals the benefits of TB to model semantic relationships of small-scale representation.

As shown in the last row of Tab. 1, GLPocket with TCB obtains further improvements in DCC and DVO under four test datasets. This is mainly because the network not only adopts TCB for global and local information modeling but also uses TB to learn the semantic correlation between patches within the proposals. Since PDBbind dataset has various compounds pairs, like protein-nucleic acid pairs etc., where the sites are relatively large. TCB in GLPocket is to remove surrounding information interference and do local refined prediction, paying more attention on the small-site detection. However, detecting larger sites needs a model with more consideration of

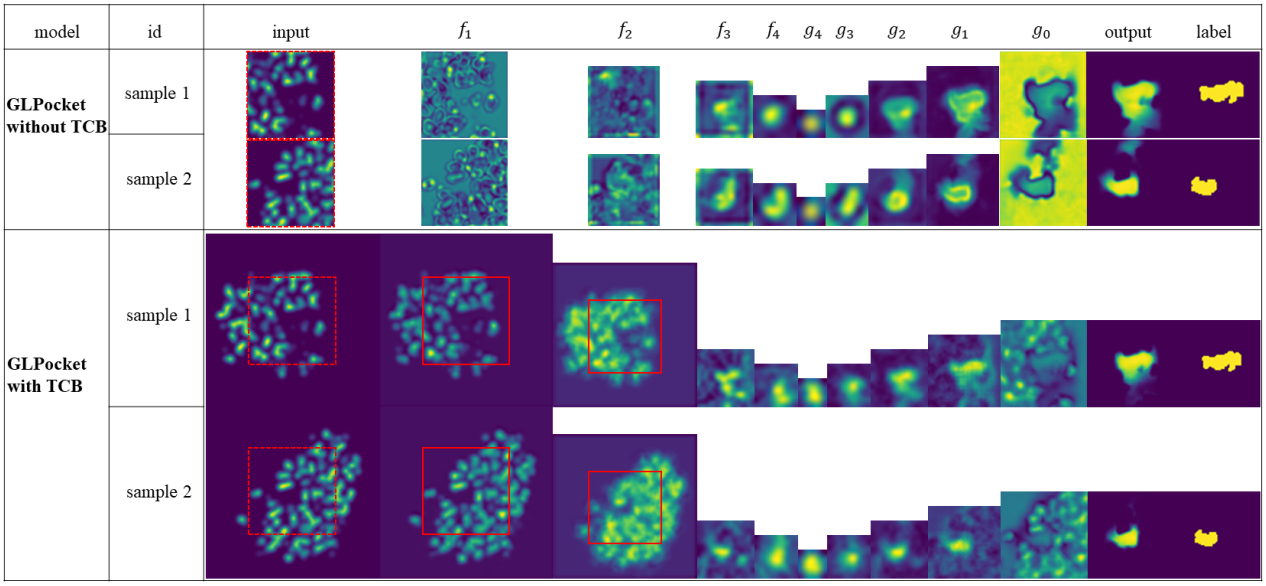

Figure 4: Visualization of cross-sections of intermediate feature maps. Features in red boxes of f_1 and f_2 are $\widetilde{f_1}$ and $\widetilde{f_2}$ respectively.

Model \ TestSet	TCB	TB	COACH420 ↑		HOLO4K ↑		SC6K ↑		PDBbind ↑		#params
			DCC	DVO	DCC	DVO	DCC	DVO	DCC	DVO	
GLPocket			85.08	54.12	82.11	53.31	84.03	50.22	63.96	36.11	25.9M
	✓		91.94	55.64	88.43	54.31	91.03	51.26	71.72	33.82	25.9M
		✓	91.13	54.86	88.78	52.57	90.90	50.90	**77.52**	**40.65**	27.5M
	✓	✓	**92.74**	**55.18**	**90.20**	**54.22**	**92.50**	**52.67**	77.14	38.51	27.5M

Table 1: We perform DCC and DVO test on different models. The bold indicates the best result.

neighborhood information in some extent. So the GLPocket without TCB delivers a superior performance on PDBbind.

3.2 Comparison to State-of-the-art Methods

We make a quantitative comparison of GLPocket with the state-of-the-art (SOTA) methods on four different test sets, as shown in Tab. 2. Note that Kalasanty [Stepniewska-Dziubinska *et al.*, 2020], DeepPocket [Aggarwal *et al.*, 2021] and RecurPocket [Li *et al.*, 2022] are the latest work to do site prediction tasks. It is found that GLPocket achieves the best performance for both DCC and DVO values. Kalasanty takes the whole protein as input, which makes it difficult to focus on small segmentation targets. DeepPocket and RecurPocket pay more attention to local proposals so their performances are better. RecurPocket builds feedback links from decoder to encoder to guide the representation learning in a recurrent and progressive manner, so it works better than DeepPocket. GLPocket takes both the global and local information into account with an appropriate trade-off, and establishes the semantic relationship between smaller patch blocks within the local proposal, so it achieves the best results.

We also report DCA on these test sets in Tab. 3. DCA Top-n mainly measures two aspects, one is the sorting ability of the prediction results, and the other is location ability for pocket. Top-n is the success rate of the first n predictions according to the priority of prediction results, where n is the number of annotated pockets for a given protein. Top-$(n+2)$ is the same. Kalasanty, DeepPocket and RecurPocket are the latest best models in DCA [Aggarwal *et al.*, 2021], and they have different ways of ranking predictions. Kalasanty sorts prediction results according to the segmentation density. It selects the first n and $n+2$ to do DCA test. DeepPocket trains a classification network and a segmentation network. The classification network is dedicated to scoring and ranking pockets predicted by Fpocket. Our work mainly solves the location and shape prediction of pockets. For a fair comparison, as in RecurPocket, models are tested in two steps. First, according to the ranking results of DeepPocket classification network, the first n and $n+2$ candidate pockets are selected to calculate candidate centers. DeepPocket, RecurPocket and GLPocket take the candidate centers as priori information of segmentation network, to predict pockets which are used to calculate DCA.

From Tab. 3 we see that GLPocket outperforms previous related works on four datasets with good generalization performance. The performance of GLPocket is 2.49% higher than RecurPocket on Top-n under COACH420 testset. Such improvements demonstrate the effectiveness of GLPocket, especially on this challenging task with such small pockets. GLPocket is slightly inferior to RecurPocket on PDBbind, but it is comparable with RecurPocket, indicating the efficiency of our methods.

TestSet / Model	COACH420 ↑		HOLO4K ↑		SC6K ↑		PDBbind ↑	
	DCC	DVO	DCC	DVO	DCC	DVO	DCC	DVO
Kalasanty	56.85	24.49	51.08	21.53	91.94	48.24	42.40	22.69
DeepPocket	85.08	54.12	83.62	51.82	84.03	50.22	63.96	36.11
RecurPocket	89.91	53.19	89.94	53.43	**92.77**	**54.22**	70.85	36.49
GLPocket	**92.74**	**55.18**	**90.20**	**54.22**	92.50	52.67	**77.14**	**38.51**

Table 2: We perform DCC and DVO test on different models. The result of RecurPocket in our paper is obtained under the condition of τ=2 using voxel-level mask. The bold indicates the best result.

TestSet / Method	COACH420 ↑		HOLO4K ↑		SC6K ↑		PDBbind ↑	
	Top-n	Top-(n + 2)	Top-n	Top-(n + 2)	Top-n	Top-(n + 2)	Top-n	Top-(n + 2)
Fpocket	35.09	51.25	36.34	51.53	23.99	37.23	19.21	43.08
Deepsite	53.07	53.07	51.65	51.67	52.94	65.41	-	-
P2Rank	68.24	75.48	70.60	80.05	62.90	75.74	*	*
Kalasanty	63.51	65.18	61.21	62.63	61.75	61.75	61.95	65.73
DeepPocket	71.53	76.87	79.79	87.56	66.39	84.33	68.89	84.56
RecurPocket	72.95	80.42	81.12	89.59	67.28	85.84	**69.71**	**85.64**
GLPocket	**75.44**	**80.43**	**81.59**	**89.62**	**67.55**	**86.19**	69.30	84.90

Table 3: DCA Top-n and Top-$n+2$ results of the state-of-the-art methods, the larger the better. The **bold** indicates the best results. n refers to the number of annotated ligand for a given protein. The symbol "-" indicates that there are no relevant results in the original paper and related papers, and the author has not published the source code or model. "*" indicates that some protein appeared in test sets are used to optimize the model parameters.

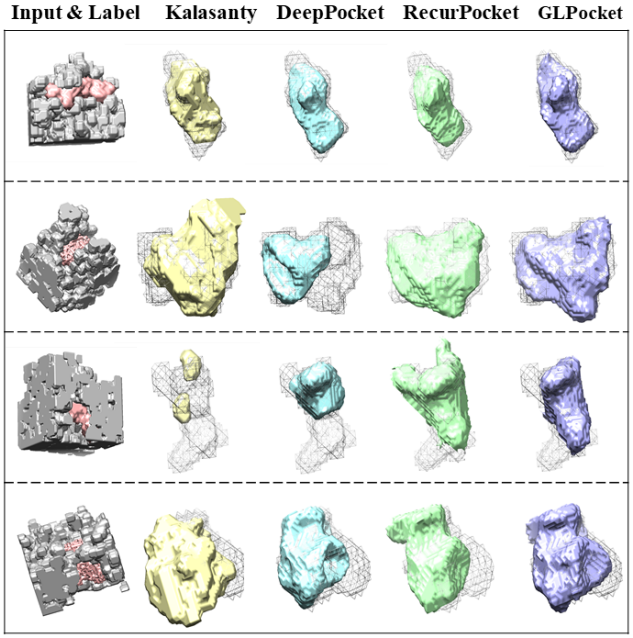

Figure 5: Visualization of four sample results of the state-of-the-art methods. The first column is partial protein and pocket, and the other columns are model prediction results. Gray grid-lines denote ground truth label.

We visualize examples of predictions in Fig. 5. The predictions by Kalasanty are often incomplete, discontinuous, and even far out-of-bounds, while those by DeepPocket and RecurPocket are more accurate and smooth. In general, the results by GLPocket are the closest to the real pockets.

4 Discussion and Conclusion

In this paper, we proposed a Lmser-based network structure, called GLPocket, to capture the multi-scale representations of objects to predict binding sites. We devised Target Cropping Block (TCB) to select interested local features from the global representation for targeted prediction. TCB makes the local region representation more discriminative by integrating the global distribution information, which helps to make the prediction more concentrated and precise. TCB also greatly reduces the volume of feature maps to be calculated by 82%, without introducing additional parameters. To fully capture the spatial dependency within local regions, we utilize Transformer Block (TB) to establish dependency relationship within proposal for enriching local context semantic information. Experiments show that GLPocket outperforms the state-of-the-art methods with a few parameters increase. This method can be extended to other detection or segmentation models easily, especially to detect or segment small targets from large-scale samples.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (grants No. 62172273), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

[Aggarwal *et al.*, 2021] 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.

[Anderson, 2003] Amy C Anderson. The process of structure-based drug design. *Chemistry & biology*, 10(9):787–797, 2003.

[Brylinski and Skolnick, 2008] 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.

[Chen *et al.*, 2011] Ke Chen, Marcin J Mizianty, Jianzhao Gao, and Lukasz Kurgan. A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. *Structure*, 19(5):613–621, 2011.

[Desaphy *et al.*, 2015] 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.

[Hendlich *et al.*, 1997] 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.

[Hernandez *et al.*, 2009] Marylens Hernandez, Dario Ghersi, and Roberto Sanchez. Sitehound-web: a server for ligand binding site identification in protein structures. *Nucleic acids research*, 37(suppl_2):W413–W416, 2009.

[Huang and Schroeder, 2006] Bingding Huang and Michael Schroeder. Ligsite csc: predicting ligand binding sites using the connolly surface and degree of conservation. *BMC structural biology*, 6(1):1–11, 2006.

[Jiménez *et al.*, 2017] 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.

[Krivák and Hoksza, 2015] 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):1–13, 2015.

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

[Li *et al.*, 2019] Peiying Li, Shikui Tu, and Lei Xu. Gan flexible lmser for super-resolution. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 756–764, 2019.

[Li *et al.*, 2022] Peiying Li, Boheng Cao, Shikui Tu, and Lei Xu. Recurpocket: Recurrent lmser network with gating mechanism for protein binding site detection. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 334–339. IEEE, 2022.

[Liu *et al.*, 2023] Yongchang Liu, Peiying Li, Shikui Tu, and Lei Xu. Refinepocket: An attention-enhanced and mask-guided deep learning approach for protein binding site prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.

[Mylonas *et al.*, 2021] 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.

[Ngan *et al.*, 2012] 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.

[Nguyen *et al.*, 2017] Nguyen, T. Quoc, Gomes, J. P. Abel, Dias, E. D. Sergio, Jorge, and A. Joaquim. Multi-gpu-based detection of protein cavities using critical points. *Future generations computer systems: FGCS*, 2017.

[Patani and LaVoie, 1996] George A Patani and Edmond J LaVoie. Bioisosterism: a rational approach in drug design. *Chemical reviews*, 96(8):3147–3176, 1996.

[Radoslav and David, 2018] Krivák Radoslav and Hoksza David. P2rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of Cheminformatics*, 10(1):39–, 2018.

[Ravindranath and Sanner, 2016] Pradeep Anand Ravindranath and Michel F Sanner. Autosite: an automated approach for pseudo-ligands prediction—from ligand-binding sites identification to predicting key ligand atoms. *Bioinformatics*, 32(20):3142–3149, 2016.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Stepniewska-Dziubinska *et al.*, 2020] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Improving detection of protein-ligand binding sites with 3d segmentation. *Scientific reports*, 10(1):1–9, 2020.

[Wang *et al.*, 2005] Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.

[Xu, 1993] Lei Xu. Least mean square error reconstruction principle for self-organizing neural-nets. *Neural networks*, 6(5):627–648, 1993.

[Xu, 2019] Lei Xu. An overview and perspectives on bidirectional intelligence: Lmser duality, double ia harmony, and causal computation. *IEEE/CAA Journal of Automatica Sinica*, 6(4):865–893, 2019.