2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

DPPBP: Dual-stream Protein-peptide Binding Sites Prediction Based on Region Detection

Yueli Yang[1], Yang Hua[1], Wenjie Zhang[1], and Xiaoning Song[1,2(✉)]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 21422, China
[2] DiTu (Suzhou) Biotechnology Co., Ltd, Suzhou, China
x.song@jiangnan.edu.cn

Abstract. Prediction of protein-peptide binding sites plays a critical role in the regulation of cellular functions and the targeted drug discovery. Recently, sequence-based prediction methods have been widely used due to their simplicity, effectiveness, and low cost of data collection. However, these methods rely on the binary classification of individual amino acids within the protein sequence, which often overlooks the dependencies between binding amino acids in the training labels. To address this issue, we propose a novel Dual-stream Protein-Peptide Binding sites Prediction method (DPPBP) based on region detection and protein language model. For the first-stream, we group successive binding sites into a single region to capture the relationships between binding amino acids and highlight the binding region of the entire sequence. Then, we use a fixed small set of learned target queries to reason about the relationships between the target regions and the global sequence information of the protein, generating the final predictions in parallel. For the second-stream, we continue to use a binary classification to discriminate each individual amino acid at a fine-grained level, and the final prediction is obtained by combining the results of both streams. Extensive experiments show that our DPPBP method outperforms the existing state-of-the-art sequence-based methods on the two benchmark datasets. Datasets and codes can be found at https://github.com/22Donkey/DPPBP.

Keywords: Protein-peptide Interaction, Binding Sites Prediction, Dual-stream Joint Inference.

1 Introduction

Protein interacts with ligands [11] such as peptides, DNA, RNA, and metal ions, playing a critical role in controlling key cellular processes such as cell metabolism, signal transduction, etc. In particular, protein-peptide interaction [25] is crucial in physiological activities such as immune responses, transcriptional regulation, cell migration and repair. Specifically, proteins protect the body from infection by binding to peptide to recognize and eliminate foreign pathogens. Besides, peptides, as part of transcription factors, modulate the transcription of specific genes, influencing cellular growth, differentiation, and stress responses. Therefore, the study of protein-peptide interaction

and their mechanisms is essential for exploring protein function [19, 33] and developing new therapeutic targets and drugs [13, 14, 15, 16, 17]. Regarding this field, traditional methods usually rely on a series of complex experimental approaches, including X-ray crystallography [20], nuclear magnetic resonance (NMR) [30], cryo-electron microscopy (cryo-EM) [3], and molecular docking [8]. These methods mimic and localize potential binding sites by resolving the 3D structure of proteins. However, these methods are limited by high costs and time-consuming experimental procedures, as well as structural resolution challenges posed by small peptide size [7] and peptide flexibility [35]. In addition, these methods typically rely on known 3D protein structures and are difficult to apply to proteins that have not yet been resolved. Therefore, sequence-based identification of protein-peptide binding sites has a broad application perspective, but still remains a challenge in the domain of biology.

Recently, with the rapid development of deep learning, various methods have been proposed to predict protein-peptide binding sites, but among them there are serious limitations for methods that require 3D structures of proteins. Obtaining high-quality 3D structures of proteins typically requires expensive experimental techniques, and the availability of structural information may be limited in cases of incomplete or unknown structures. Although AlphaFold [2] has achieved remarkable success in the domain of protein structure prediction, the predicted structural data can introduce misinformation into the prediction of binding sites [10] and then cause errors. Therefore, sequence-based methods continue to be commonly applied thanks to their simplicity and efficiency.

Sequence-based protein-peptide binding sites prediction approaches are based on a binary classification of individual amino acids within the protein sequence and offer high computational efficiency, allowing for the rapid processing of large-scale protein data without the need for detailed sequence information. However, these methods suffer from several fundamental limitations: **First**, binding sites are typically composed of spatially adjacent amino acid residues that manifest as either dispersed or contiguous regions in the primary structure and complex interactive relationships in the tertiary structure. Current methods inadequately capture the local dependencies and global sequence features among these amino acids. They overlook the semantic relationships between amino acids and the interdependencies between them within binding regions. This results in an inability to accurately identify the overall integrity of binding sites and the synergistic effects between regions, which is particularly inadequate for long and complex binding sites. **Second**, protein sequences often exhibit a considerable imbalance between binding sites and non-binding sites, with binding sites making up only a small fraction of the entire sequence. This imbalance can cause models to bias their predictions toward non-binding sites during training, thereby hindering the accurate identification of binding sites.

To address the above limitations, we propose a novel dual-stream joint inference method, called DPPBP, based on SPN [26]. As shown in Fig. 1, the proposed method is inspired by detection methods in computer vision and natural language processing [34]. **First**, we group successive binding sites into a single region, allowing the network to learn the relationships between the binding regions and the global sequence information of the protein, and automatically extract potential binding regions in the first-

stream. A bipartite matching loss function, which directly optimizes for the best match, is used to mitigate the effects of data imbalance, especially for binding regions with lower representation. **Second**, we continue to use binary classification in the second-stream, focusing on discriminating each individual amino acid, and the final prediction is derived by taking the union of the outputs from both streams. The combination of contrastive loss and cross-entropy loss is used to balance the contributions of different classes in the binary classification task, ensuring that the model is appropriately optimized across all classes. Extensive experiments and ablation studies prove that our proposed method greatly improves the performance of binding sites prediction method, and each innovation is valuable to the whole framework.

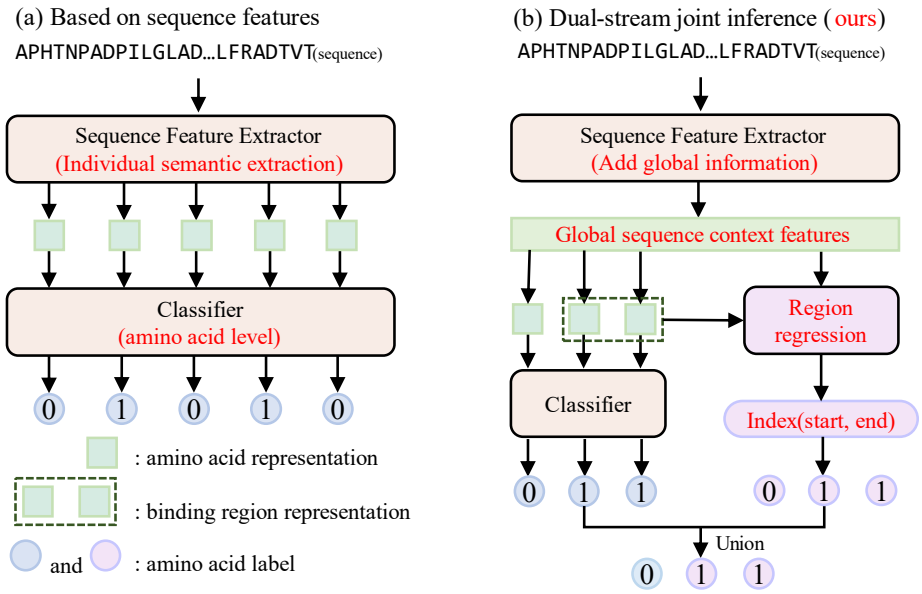

Fig. 1. (a) shows the method based on protein sequence features. (b) shows our proposed dual-stream joint inference model.

2 Related Work

Sequence-based binding sites prediction methods have been widely applied due to their simplicity and efficiency. First, Taherzadeh et al. proposed the SPRINT-Seq [28] model, which encodes protein sequences as one-hot vector features and combines them with support vector machines (SVMs) for prediction. Next, Zhao et al. developed the PepBind [37] model, which further expanded the feature space of protein sequences by introducing intrinsic disorder features, revealing the relationship between protein-peptide binding and inherent disorder. Subsequently, Wardah et al. proposed a two-step Visual [32] method. The first step extracts relevant features from protein sequences. The second step encodes each residue and its neighbors into an image-like

representation using a sliding window and predicts binding residues using a convolutional neural network (CNN). Additionally, Abdin et al. developed the PepNN-Seq [1] method based on the reciprocal attention mechanism that concurrently updates peptide and protein representations, better reflecting the conformational changes occurring during binding. Similarly, the PepCA [18] method proposed by Huang et al. achieved high-precision binding sites prediction by encoding protein sequences using the ESM-2 [23] pre-training model and updating the encoding using a multi-input coattention module. Finally, the PepBCL [31] model proposed by Wang et al. combined the protBert [4] model with contrastive learning methods, providing an end-to-end solution that further optimizes the pre-trained embeddings of protein sequences, thereby improving the accuracy of protein-peptide binding sites prediction.

3 Materials and Methods

3.1 The Benchmarking Datasets

To objectively evaluate and contrast the performance of our proposed method with current methods, we selected two benchmarking datasets, denoted as Dataset 1 and Dataset 2, from the BioLip [36] database. These datasets, with a protein residue-level positive-to-negative ratio of 16,749:290,943 (≈1:17.4), are widely adopted for training and evaluating protein-peptide binding sites prediction models. A brief summary of these is given in Table 1, and their detailed descriptions follow.

Dataset 1 was proposed by Taherzadeh et al. during the development of the structure-based binding sites prediction model, SPRINT-Str [29]. They randomly divided the protein sequences with peptide binding into two subsets: a training set (labeled as TR1154) and an independent test set (labeled as TE125). The training set accounts for 90% of the data, and the test set accounts for 10%. The TR1154 training set contains 1,154 protein sequences, which include 9,010 peptide binding regions, with the remaining regions being background non-binding regions. **Dataset 2** was proposed by Zhao et al. during the development of the sequence-based binding sites prediction model, PepBind [37]. They randomly divided 1,279 peptide-binding protein sequences into two equally sized subsets, creating a training set (labeled as TR640) and a test set (labeled as TE639). The TR640 training set contains 640 protein sequences, which include 4,970 peptide binding regions, with the remaining regions being background non-binding regions.

Table 1. Statistics of the benchmarking datasets. Consecutive binding residues in a sequence are a binding region.

	Dataset 1		Dataset 2	
	Train	Test	Train	Test
Proteins	1,154	125	640	639
Binding regions	9,010	1,039	4,970	5,079
Binding residues	15,030	1,719	8,259	8,490
Non-binding residues	261,792	29,151	149,103	141,840

3.2 The Proposed Method

The overall framework of the DPPBP method proposed in this study is shown in Fig. 2. Both streams use the protein pre-trained model ProtBert for representation. The first-stream uses a transformer-based non-autoregressive decoder [12, 38] to decode the start and end indices of each binding region, while the second-stream uses a series of linear layers to perform a binary classification on each amino acid to determine whether it is a binding residue. Finally, the overall output is obtained by taking the union of the predictions from both streams. The details of both are presented in the subsequent sections.

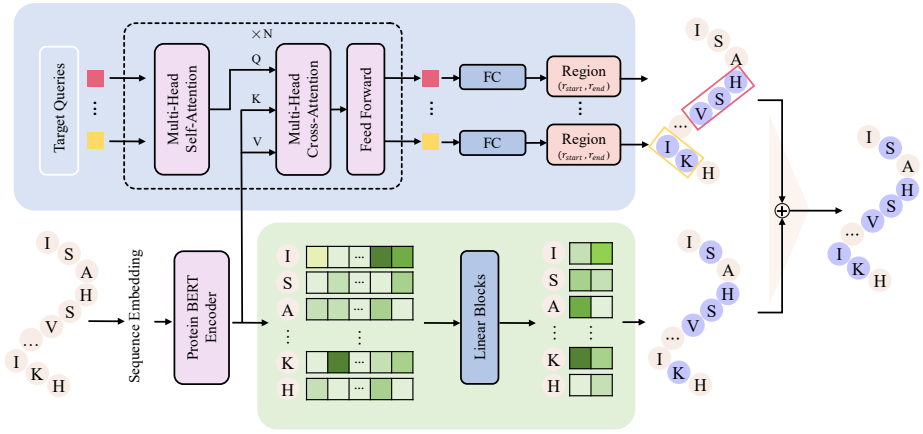

Fig. 2. The overall pipeline of DPPBP, where the top part is the detection process of binding regions for the first-stream, and the bottom part is the binary classification process for each amino acid for the second-stream. Both streams share the encoder ProtBert.

First-stream: prediction of binding regions. For the first-stream, we predict binding regions by grouping successive binding amino acids into a single region. The task is formulated as a set prediction problem, with the set generated directly by a transformer-based non-autoregressive decoder. The set contains the start and end indices of the binding region as part of the region information. Unlike sequence-to-sequence

autoregressive methods, the non-autoregressive decoder avoids learning the generation order of the set, thus eliminating the dependence on sequence order. Additionally, it fully uses bidirectional contextual information, rather than being constrained by unidirectional generation, which allows for more efficient extraction of set-level information.

Non-autoregressive decoder. Before decoding begins, we initialize the input with m learnable embeddings called target queries, similar to those used in SPN. These target queries are shared across all input sequences, meaning that the same target queries are used for initialization in each protein sequence. Therefore, the model first needs to know the size of the target set, which means that the $P_L(n|S)$ in Equation (1) needs to be modeled first. The size of the target set refers to the number of binding regions in each protein sequence in the dataset. To simplify, we allow the non-autoregressive decoder to generate m fixed target predictions for each input sequence, where m is set larger than the maximum number of binding regions in any protein sequence in the dataset. This removes the need for the decoder to explicitly model the target set size, instead addressing it indirectly by generating a fixed number of target predictions.

By design, the non-autoregressive decoder can be modeled directly based on Equation (1). Specifically, the core objective of this task is to identify all potential position indices from a given protein sequence and organize them into a set of regions. Formally, given an input sequence S, the conditional probability of the target set $T = \{(r_1^{start}, r_1^{end}), \dots, (r_n^{start}, r_n^{end})\}$ can be expressed as:

$$P(T|S;\mu) = P_L(n|S)\prod_{i=1}^{n} p\left(T_i\middle|S, T_{j\neq i};\mu\right) \tag{1}$$

where $T_i = (r_i^{start}, r_i^{end})$, r_i^{start} and r_i^{end} are the start and end indices of the binding region, respectively, and $p\left(T_i\middle|S, T_{j\neq i};\mu\right)$ is the conditional probability of generating each region T_i, which is not only dependent on the protein sequence S, but also on the relationships with other regions $T_{j\neq i}$ in the set. μ is a learnable parameter.

The non-autoregressive decoder consists of N identical transformer decoder blocks. Each transformer module includes a multi-head self-attention layer to capture the relationships between the binding regions, and a multi-head cross-attention layer to capture the relationships between the binding regions and the protein sequence embedding information produced by the ProtBert encoder. During the decoding process, the m target queries are transformed into m output embedding matrices, denoted as $P_d \in \mathbb{R}^{m\times d}$. Subsequently, these output embeddings are decoded one by one through the fully connected layer (FC) into the start and end indices of the binding regions, generating the final predictions m. In particular, the number of target regions in the sequence is not necessarily equal to m, as some query vectors may predict the background or exhibit duplicate predictions. Specifically, given an embedding vector $p_d \in \mathbb{R}^d$ from the output embedding matrix P_d, we predict the start and end indices of the binding regions through two independent classifiers. This process is described by the following equations:

$$p^{start} = Softmax(a_1^T \tanh(A_1 p_d + A_2 P_e)) \quad (2)$$

$$p^{end} = Softmax(a_2^T \tanh(A_3 p_d + A_4 P_e)) \quad (3)$$

where $P_e \in \mathbb{R}^{l \times d}$ is the protein sequence embedding produced by the ProtBert encoder, $A_1, A_2, A_3, A_4 \in \mathbb{R}^{d \times d}$ and $a_1, a_2 \in \mathbb{R}^d$ are learnable parameters.

Bipartite matching loss function. For the first-stream, we use a bipartite matching loss [5] to avoid excessive reliance on candidate regions, while considering the specific characteristics of the predicted region sets. Through optimal matching, this loss function helps the model better align predictions with the ground truth during training. The process consists of two main steps: finding an optimal match and calculating the loss. To find the optimal match, we first compute the matching cost based on the similarity between predicted and ground truth pairs. We calculate the cost using a weighted average, and then use the Hungarian algorithm to determine the best matching pairs.

We denote the set of m predicted pairs as $\tilde{T} = \{\tilde{T}_1, \tilde{T}_2 \dots, \tilde{T}_m\}$ and the set of ground truth pairs as $T = \{T_1, T_2, \dots, T_n\}$, where each T_i is the i-th pair $T_i = (r_i^{start}, r_i^{end})$, Note that, $m \geq n$, typically means the number of predicted pairs is greater than or equal to the number of ground truth pairs. To ensure that each ground truth pair has a matching predicted pair, we pad the set of ground truth pairs with an empty set $\emptyset$, resulting in a new set $T = \{T_1, T_2, \dots, T_m\}$, thereby making the size of the set consistent with the predicted set.

To achieve optimal matching, we use a permutation optimization method. Specifically, we calculate the matching cost for each pair of ground truth and predicted pairs and then minimize the total matching cost to obtain the optimal match. The optimization objective can be expressed as follows:

$$\theta^* = \underset{\theta \in \prod(m)}{\operatorname{argmin}} \sum_{i=1}^{m} Cost\left(T_i, \tilde{T}_{\theta(i)}\right) \quad (4)$$

where $\theta \in \prod(m)$ is all possible matching permutations, $\prod(m)$ is the permutation space of length m. $Cost\left(T_i, \tilde{T}_{\theta(i)}\right)$ is the matching cost between the ground truth pair T_i and the predicted pair $\tilde{T}_{\theta(i)}$.

The matching cost $Cost\left(T_i, \tilde{T}_{\theta(i)}\right)$ is defined as:

$$Cost\left(T_i, \tilde{T}_{\theta(i)}\right) = -P^r_{\theta(i)}(r_i) \quad (5)$$

where $P^r_{\theta(i)}(r_i)$ is the matching of the region span.

The region span matching cost considers the matching of both the start and end indices of the region:

$$P^r_{\theta(i)}(r_i) = P^{start}_{\theta(i)}(r_i^{start}) + P^{end}_{\theta(i)}\left(r_i^{end}\right) \quad (6)$$

The second step is to calculate the loss function, which is the total of the matching costs between all ground truth and predicted pairs, resulting in the final loss value. This is defined as:

$$\mathcal{L}(T,\tilde{T}) = -\sum_{i=1}^{m}[\log p_{\theta^*(i)}^{start}(r_i^{start}) + \log p_{\theta^*(i)}^{end}(r_i^{end})] \tag{7}$$

Second-stream: binding residues classifier. The second-stream performs the common task of binary classification of each residue in the protein sequence.

Linear binary classifier. The input protein sequence S is first encoded by ProtBert [4] to produce a feature matrix $P_e \in \mathbb{R}^{l\times d}$, where each row vector $e_{s_i} \in \mathbb{R}^d$ is the high-dimensional embedding of each amino acid residue. Subsequently, these high-dimensional embeddings are processed through a series of linear layers, transforming them into two-dimensional embedding vectors:

$$z_{s_i} = W_l e_{s_i} + b \tag{8}$$

where $W_l \in \mathbb{R}^{2\times d}$ is the weight matrix of the linear blocks, $b \in \mathbb{R}^d$ is the bias term, $z_{s_i} \in \mathbb{R}^2$ is the two-dimensional vector representing each amino acid, $z_{s_i} = [\begin{smallmatrix} z_{s_i,0} \\ z_{s_i,1} \end{smallmatrix}]$.

Finally, the Softmax function is used to each amino acid's output z_{s_i} to calculate the probabilities p_0 and p_1 for each amino acid:

$$p_{s_i,0} = Softmax(z_{s_i}) = \frac{\exp(z_{s_i,0})}{\exp(z_{s_i,0}) + \exp(z_{s_i,1})} \tag{9}$$

$$p_{s_i,1} = Softmax(z_{s_i}) = \frac{\exp(z_{s_i,1})}{\exp(z_{s_i,0}) + \exp(z_{s_i,1})} \tag{10}$$

where $z_{s_i,0}$ is the score for class 0 (non-binding sites) for the i-th amino acid, $z_{s_i,1}$ is the score for class 1 (binding sites) for the i-th amino acid, $p_{s_i,0}$ and $p_{s_i,1}$ are the probabilities of the i-th amino acid belonging to class 0 and class 1, respectively.

Contrastive loss and cross-entropy loss function. For the second-stream, to address the issue of class imbalance, we use a loss function that combines contrastive loss and the standard binary cross-entropy loss. Particularly, the contrastive loss is defined as:

$$\mathcal{L}_1(e_1,e_2,b) = \frac{1}{2}(1-b)\cdot D(e_1,e_2)^2 + \frac{1}{2}b\cdot\left(D_{max} - D(e_1,e_2)\right)^3 \tag{11}$$

$$D(e_1,e_2) = 1 - cos < e_1, e_2 > \tag{12}$$

where e_1 and e_2 are two different residues. $D(e_1,e_2)$ is the distance metric between e_1 and e_2 based on cosine similarity.

The value of $D(e_1, e_2)$ ranges from 0 to 2, D_{max} is the maximum value of $D(e_1, e_2)$. $b = 0$ indicates that the pair of amino acids belong to the same class. $b = 1$ indicates that the pair of amino acids belong to different classes, meaning one is a binding residue and the other is not. To alleviate the class imbalance issue, we set two different weights. When $b = 1$, the loss function $\mathcal{L}_1 = \frac{1}{2} b \cdot \left(D_{max} - D(e_1, e_2)\right)^3$ attempts to maximize the distance between residues of the different class. When $b = 0$, the loss function $\mathcal{L}_1 = \frac{1}{2} \cdot D(e_1, e_2)^2$ attempts to reduce the distance between the same class.

The binary cross-entropy loss function is described as:

$$\mathcal{L}_{CE}(p_1, y) = -y log p_1 - (1 - y) \log(1 - p_1) \tag{13}$$

where p_1 is the probability predicted by the model that the residue is a positive sample (i.e., binding sites), and $1 - p_1$ is the probability that the residue is a negative sample (i.e., non-binding sites). y is the true label.

Finally, the total loss for a batch with N residues is:

$$\mathcal{L} = \sum_{i=1}^{N/2} \mathcal{L}1\left(e_i, e_{N/2+i}, b\right) + \sum_{i=1}^{N} \mathcal{L}_{CE}\left(p_{1,i}, y_i\right) \tag{14}$$

where e_i and $e_{N/2+i}$ are paired residues, and $p_{1,i}$ and y_i are the predicted probability of the i-th residue being in the positive class and the true label, respectively.

3.3 Implementation Details

In our proposed method, to ensure consistency and collaboration between the two streams, the parameters of the ProtBert encoder are shared with a very small learning rate of 1e-5. During training, both streams use the same optimizer, AdamW. The two streams are then trained using different decoding models and loss functions to achieve optimal performance. The first-stream uses a non-autoregressive decoder and a bipartite matching loss function, trained for 50 epochs with an initial learning rate of 2e-5. The second-stream uses linear classification layers, contrastive loss and cross-entropy loss function, trained for 10 epochs. Finally, the best model weights from both streams are used for the test set, and the union of their predictions is taken as the final evaluation result. All experiments were conducted on an NVIDIA GeForce A30 GPU.

4 Results

In this section, we will compare our method with current methods on the two benchmark datasets. Next, we report the results of an ablation study to compare the performance of the single-stream and dual-stream approaches. Finally, to validate the interpretability of our approach, we visualize and analysis the experimental results, including the results predicted by the PepBCL and the proposed method.

4.1 Comparison with State-of-the-Art Methods

To comprehensively evaluate the performance of the method proposed in this study and compare it with existing state-of-the-art methods, we adopt the same evaluation metrics used in previous studies, including Recall, Specificity, Precision, AUC (Area Under the ROC Curve), and MCC (Matthews Correlation Coefficient).

Our method is compared to ten existing approaches on the TR1154 training set and the TE125 test set. These methods include Pepsite [24], Peptimap [21], SPRINT-Seq [28], SPRINT-Str [29], PepBind [37], PepNN-Seq [1], PepNN-Struct[1], PepBCL [31], PepCNN [6] and PepPFN [22]. Five of these methods are add structural features (i.e., Pepsite, Peptimap, SPRINT-Str, PepNN-Struct, PepCNN), while the remaining methods are sequence-based prediction methods. We also perform a comparison on dataset 2, consisting of TR640 training set and TE639 test set. In this case, we compare our method with five methods: PepNN-Seq, PepNN-Struct, PepBCL, PepCNN and PepPFN, where PepNN-Struct and PepCNN also add structural features. The experimental comparison results on the two datasets are provided in Table 2 and Table 3, respectively, where the results of the existing methods are directly quoted from the related literature.

As shown in Table 2, our method demonstrates superior performance on the TE125 test set across both Recall and MCC metrics, even outperforming methods that incorporate structural features. In terms of AUC, it also achieves the highest performance level among all sequence-based methods. Specifically, when compared with the structure-enhanced PepCNN method, our approach shows improvements of 10.9% and 5.4% in Recall and MCC metrics, respectively. Similarly, compared to the sequence-based PepPFN method, our method exhibits superior performance with increases of 16.8% in Recall, 0.8% in AUC, and 8.2% in MCC. Further evaluation on the TE639 test set (Table 3) reveals that our method achieves the best results in Recall, Precision, and MCC metrics among all evaluated methods, including both sequence-based and structure-based approaches.

Table 2. A comparison with state-of-the-art methods on the TE125 test set, marked * are methods with added structural features.

Methods	Recall	Specificity	Precision	AUC	MCC
Pepsite*	0.180	0.970	-	0.610	0.200
Peptimap*	0.320	0.950	-	0.630	0.270
SPRINT-Seq	0.210	0.960	-	0.680	0.200
SPRINT-Str*	0.240	0.980	-	0.780	0.290
PepBind	0.344	-	0.469	0.793	0.372
PepNN-Seq	-	-	-	0.805	0.278
PepNN-Struct*	-	-	-	0.841	0.321
PepBCL	0.315	0.984	0.540	0.815	0.385
PepCNN*	0.254	0.988	0.550	**0.843**	0.350
PepPFN	0.195	**0.992**	**0.600**	0.813	0.322
DPPBP(ours)	**0.363**	0.980	0.513	0.821	**0.404**

Table 3. A comparison with state-of-the-art methods on the TE639 test set, marked * are methods with added structural features.

Methods	Recall	Specificity	Precision	AUC	MCC
PepNN-Seq	-	-	-	0.792	0.251
PepNN-Struct*	-	-	-	**0.838**	0.301
PepBCL	0.252	0.983	0.470	0.804	0.312
PepCNN*	0.217	0.986	0.479	0.826	0.297
PepPFN	0.127	**0.996**	**0.680**	0.813	0.307
DPPBP(ours)	**0.275**	0.983	0.493	0.793	**0.341**

4.2 Ablation Studies

In the ablation study of this research, we performed an in-depth analysis of the performance differences between the single-stream and dual-stream models. Specifically, we trained each stream on the TR1154 training set and evaluated their performance on the TE125 test set. Similarly, each stream was trained on the TR640 dataset and evaluated on the TE639 test dataset.

As shown in Table 4. The ablation study results show that on the TE125 and TE639 test sets, the MCC of the first-stream model is relatively lower, with values of 34.3% and 20.1%, respectively. In contrast, the second-stream model shows a certain improvement, achieving 36.7% and 29.7%, respectively. Overall, the dual-stream joint inference method effectively leverages the strengths of each stream to achieve a balanced performance across various evaluation metrics. The first-stream focuses on locating binding regions with a high level of abstraction, while the second-stream focuses more on fine-grained amino acid classification. By taking the union of their predictions, the method improves the predictive performance of the results.

Table 4. Ablation study results on the TE125 and TE639 test set.

Datasets	Stream	Recall	Specificity	Precision	MCC
TE125	Stream 1	0.199	**0.993**	**0.654**	0.343
	Stream 2	0.303	0.983	0.510	0.367
	DPPBP	**0.363**	0.980	0.513	**0.404**
TE639	Stream 1	0.181	0.976	0.307	0.201
	Stream 2	0.270	0.976	0.401	0.297
	DPPBP	**0.275**	**0.983**	**0.493**	**0.341**

4.3 Analysis of Visualization Results

To further evaluate the prediction performance and effectiveness of our model, three protein sequences were randomly chosen from the TE125 test set (Protein IDs: 1dpu, 1uj0, and 2bbu) to predict their binding sites. To visually demonstrate the accuracy of

the predictions and the model's ability to recognize binding sites, we first retrieved the corresponding 3D structures of these sequences from the PDB database [27] and then visualized these structures using PyMOL [9] software, as shown in Fig. 3. Each protein is represented by three rows of sequence-structure images: the first row (Fig. 3 A, B, C) shows the experimentally obtained true binding sites, the second row (Fig. 3 D, E, F) shows the results predicted by the existing method PepBCL, and the third row (Fig. 3 G, H, I) shows the results predicted by our method DPPBP. Each image includes the protein sequence and its corresponding 3D structure, with the binding sites highlighted using the same color in both the sequence and structure.

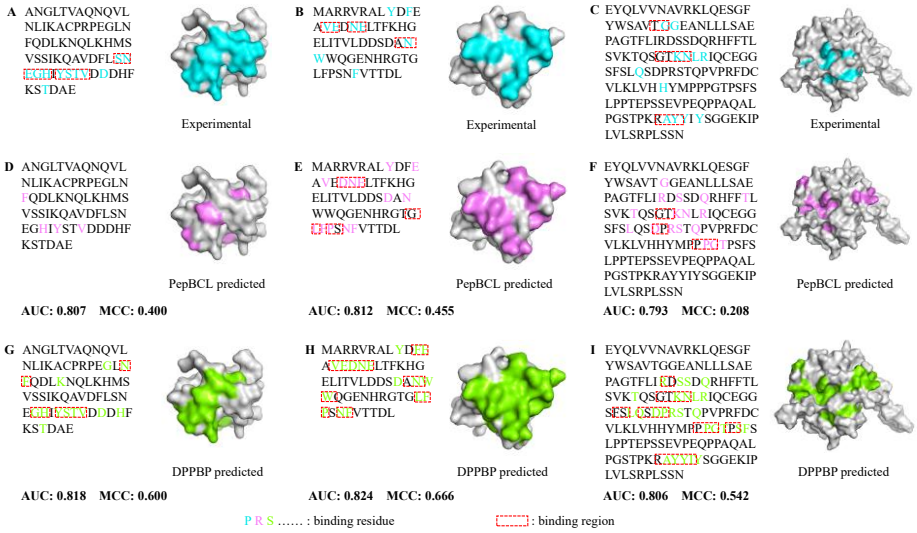

Fig. 3. Comparison of protein binding regions visualization results. (Gray: Non-binding regions, Blue: Experimentally validated binding regions, Purple: PepBCL predicted regions, Green: Our method's predicted regions.)

Taking the 1dpu protein as an example (Fig. 3 A, D, G), we observe that while the PepBCL method accurately captures most of the true binding sites, it fails to account for the dependencies between binding regions, leading to the separation of contiguous binding regions. In contrast, our method not only predicts binding sites that closely match the true binding sites, but also ensures that the binding regions are more continuous, with higher precision at the boundaries. Despite variations in sequence length and features among different proteins, our model can adaptively capture local dependencies and global features, enabling accurate prediction of binding sites.

5 Conclusion and Future Works

We propose a dual-stream joint inference method (DPPBP), based on protein sequence for binding sites prediction. The approach introduces the concept of region detection and incorporates a dual-stream joint inference strategy to enhance prediction accuracy. Experimental results demonstrate that this dual-stream joint inference design enables the model not only to better capture the relationships between binding amino acids and binding regions, but also to effectively incorporate the global sequence information, thereby improving overall predictive performance. Despite the strong performance of DPPBP, several limitations remain. The dual-stream joint inference requires computation in both streams, which increases the complexity and optimization challenges of the model, necessitating a more refined training process. Future studies may concentrate on optimizing the model's computational efficiency, addressing error accumulation, and exploring ways to extend DPPBP to predict a broader range of binding site types. These advances would contribute to the progress and application of drug discovery and bioinformatics.

Acknowledgments. This work was supported by the National Key R\&D Program of China (2023YFF1105102, 2023YFF1105105), the National Natural Science Foundation of China (Grant NO. 62106089, 62336004), the Major Project of the National Social Science Foundation of China (No. 21\&ZD166) and the Natural Science Foundation of Jiangsu Province (No. BK20221535).

References

1. Abdin, O., Wen, H., Kim, P.M.: Sequence and structure based deep learning models for the identification of peptide binding sites. Advances in Neural Information Processing Systems 33 (2020)
2. Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A.J., Bambrick, J., et al.: Accurate structure prediction of biomolecular interactions with alphafold 3. Nature pp. 1–3 (2024)
3. Atherton, J., Moores, C.A.: Cryo-em of kinesin-binding protein: challenges and opportunities from protein-surface interactions. Acta Crystallographica Section D: Structural Biology 77(4), 411–423 (2021)
4. Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., Linial, M.: Proteinbert: a universal deep-learning model of protein sequence and function. Bioinformatics 38(8), 2102–2110 (2022)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
6. Chandra, A., Sharma, A., Dehzangi, I., Tsunoda, T., Sattar, A.: Pepcnn deep learning tool for predicting peptide binding residues in proteins using sequence, structural, and language model features. Scientific reports 13(1), 20882 (2023)
7. Chen, S., Bertoldo, D., Angelini, A., Pojer, F., Heinis, C.: Peptide ligands stabilized by small molecules. Angewandte Chemie International Edition 53(6), 1602–1606 (2014)

8. Ciemny, M., Kurcinski, M., Kamel, K., Kolinski, A., Alam, N., Schueler-Furman, O., Kmiecik, S.: Protein–peptide docking: opportunities and challenges. Drug discovery today 23(8), 1530–1537 (2018)
9. DeLano, W.L., et al.: Pymol: An open-source molecular graphics tool. CCP4 Newsl. Protein Crystallogr 40(1), 82–92 (2002)
10. Deng, B., Hua, Y., Zhang, W., Song, X., Wu, X.j.: Drivpocket: A dual-stream rotation invariance in feature sampling and voxel fusion approach for protein binding site prediction. In: International Conference on Pattern Recognition. pp. 203–219. Springer (2025)
11. Dhakal, A., McKay, C., Tanner, J.J., Cheng, J.: Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. Briefings in Bioinformatics 23(1), bbab476 (2022)
12. Guo, J., Tan, X., He, D., Qin, T., Xu, L., Liu, T.Y.: Non-autoregressive neural machine translation with enhanced decoder input. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 3723–3730 (2019)
13. Hua, Y., Li, J., Feng, Z., Song, X., Sun, J., Yu, D.: Protein drug interaction prediction based on attention feature fusion. J Comput Res Develop 59(9), 2051–65 (2022)
14. Hua, Y., Feng, Z., Song, X., Li, H., Xu, T., Wu, X.J., Yu, D.J.: Apmg: 3d molecule generation driven by atomic chemical properties. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2024)
15. Hua, Y., Feng, Z., Song, X., Wu, X.J., Kittler, J.: Mmdg-dti: Drug–target interaction prediction via multimodal feature fusion and domain generalization. Pattern Recognition 157, 110887 (2025)
16. Hua, Y., Song, X., Feng, Z., Wu, X.J., Kittler, J., Yu, D.J.: Cpinformer for efficient and robust compound-protein interaction prediction. IEEE/ACM transactions on computational biology and bioinformatics 20(1), 285–296 (2022)
17. Hua, Y., Song, X., Feng, Z., Wu, X.: Mfr-dta: a multi-functional and robust model for predicting drug–target binding affinity and region. Bioinformatics 39(2), btad056 (2023)
18. Huang, J., Li, W., Xiao, B., Zhao, C., Zheng, H., Li, Y., Wang, J.: Pepca: Unveiling protein-peptide interaction sites with a multi-input neural network model. Iscience 27(10) (2024)
19. Kulmanov, M., Hoehndorf, R.: Deepgozero: improving protein function prediction from sequence and zero-shot learning based on ontology axioms. Bioinformatics 38(Supplement_1), i238–i245 (2022)
20. Ladd, M.F.C., Palmer, R.A., Palmer, R.A.: Structure determination by X-ray crystallography, vol. 233. Springer (1977)
21. Lavi, A., Ngan, C.H., Movshovitz-Attias, D., Bohnuud, T., Yueh, C., Beglov, D., Schueler-Furman, O., Kozakov, D.: Detection of peptide-binding sites on protein surfaces: The first step toward the modeling and targeting of peptide-mediated interactions. Proteins: Structure, Function, and Bioinformatics 81(12), 2096–2105 (2013)
22. Li, X., Cao, B., Ding, H., Kang, N., Song, T.: Peppfn: protein-peptide binding residues prediction via pre-trained module-based fourier network. In: 2024 IEEE Conference on Artificial Intelligence (CAI). pp. 1075–1080. IEEE (2024)
23. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al.: Evolutionary-scale prediction of atomic-level protein structure with a language model. Science 379(6637), 1123–1130 (2023)
24. Petsalaki, E., Stark, A., García-Urdiales, E., Russell, R.B.: Accurate prediction of peptide binding sites on protein surfaces. PLoS computational biology 5(3), e1000335 (2009)
25. Shanker, S., Sanner, M.F.: Predicting protein–peptide interactions: benchmarking deep learning techniques and a comparison with focused docking. Journal of Chemical Information and Modeling 63(10), 3158–3170 (2023)

26. Sui, D., Zeng, X., Chen, Y., Liu, K., Zhao, J.: Joint entity and relation extraction with set prediction networks. IEEE Transactions on Neural Networks and Learning Systems (2023)
27. Sussman, J.L., Lin, D., Jiang, J., Manning, N.O., Prilusky, J., Ritter, O., Abola, E.E.: Protein data bank (pdb): database of three-dimensional structural information of biological macromolecules. Acta Crystallographica Section D: Biological Crystallography 54(6), 1078–1084 (1998)
28. Taherzadeh, G., Yang, Y., Zhang, T., Liew, A.W.C., Zhou, Y.: Sequence-based prediction of protein–peptide binding sites using support vector machine. Journal of computational chemistry 37(13), 1223–1229 (2016)
29. Taherzadeh, G., Zhou, Y., Liew, A.W.C., Yang, Y.: Structure-based prediction of protein–peptide binding regions using random forest. Bioinformatics 34(3), 477–484 (2018)
30. Tugarinov, V., Ceccon, A., Clore, G.M.: Nmr methods for exploring 'dark'states in ligand binding and protein-protein interactions. Progress in nuclear magnetic resonance spectroscopy 128, 1–24 (2022)
31. Wang, R., Jin, J., Zou, Q., Nakai, K., Wei, L.: Predicting protein–peptide binding residues via interpretable deep learning. Bioinformatics 38(13), 3351–3360 (2022)
32. Wardah, W., Dehzangi, A., Taherzadeh, G., Rashid, M.A., Khan, M.G., Tsunoda, T., Sharma, A.: Predicting protein-peptide binding sites with a deep convolutional neural network. Journal of Theoretical Biology 496, 110278 (2020)
33. Watson, J.L., Juergens, D., Bennett, N.R., Trippe, B.L., Yim, J., Eisenach, H.E., Ahern, W., Borst, A.J., Ragotte, R.J., Milles, L.F., et al.: De novo design of protein structure and function with rfdiffusion. Nature 620(7976), 1089–1100 (2023)
34. Xiao, X., Wang, W., Xie, J., Zhu, L., Chen, G., Li, Z., Wang, T., Xu, M.: Hgtdp-dta: Hybrid graph-transformer with dynamic prompt for drug-target binding affinity prediction. arXiv preprint arXiv:2406.17697 (2024)
35. Xu, X., Zou, X.: Predicting protein–peptide complex structures by accounting for peptide flexibility and the physicochemical environment. Journal of chemical information and modeling 62(1), 27–39 (2021)
36. Yang, J., Roy, A., Zhang, Y.: Biolip: a semi-manually curated database for biologically relevant ligand–protein interactions. Nucleic acids research 41(D1), D1096–D1103 (2012)
37. Zhao, Z., Peng, Z., Yang, J.: Improving sequence-based prediction of protein–peptide binding residues by introducing intrinsic disorder and a consensus method. Journal of Chemical Information and Modeling 58(7), 1459–1468 (2018)
38. Zhu, M., Wang, J., Yan, C.: Non-autoregressive neural machine translation with consistency regularization optimized variational framework. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 607–617 (2022)