


## ORIGINAL RESEARCH OPEN ACCESS

# ACP-DPE: A Dual-Channel Deep Learning Model for Anticancer Peptide Prediction

Guohua Huang<sup>1,2</sup> | Yujie Cao<sup>1</sup> | Qi Dai<sup>3</sup> | Weihong Chen<sup>2</sup> 

<sup>1</sup>College of Information Science and Engineering, Shaoyang University, Shaoyang, China | <sup>2</sup>Hunan Provincial Key Laboratory of Finance & Economics Big Data Science and Technology, Hunan University of Finance and Economics, Changsha, China | <sup>3</sup>College of Life Science and Medicine, Zhejiang Sci-Tech University, Hangzhou, China

**Correspondence:** Weihong Chen ([chenweihong@hufe.edu.cn](mailto:chenweihong@hufe.edu.cn))

**Received:** 27 December 2024 | **Revised:** 13 February 2025 | **Accepted:** 20 February 2025

**Handling Editor:** Hao Wu

**Funding:** This work was supported in part by the Special Support Plan Project for High level Talents in Zhejiang Province (Grant No. 2021R52019), by the Scientific Research Fund of Hunan Provincial Education Department (Grant Nos. 24A0694 and 24A0701), and by the Shaoyang University Postgraduate Scientific Research Innovation Project (Grant No. CX2023SY046).

**Keywords:** bioinformatics | biology | drugs

## ABSTRACT

Cancer is a serious and complex disease caused by uncontrolled cell growth and is becoming one of the leading causes of death worldwide. Anticancer peptides (ACPs), as a bioactive peptide with lower toxicity, emerge as a promising means of effectively treating cancer. Identifying ACPs is challenging due to the limitation of experimental conditions. To address this, we proposed a dual-channel-based deep learning method, termed ACP-DPE, for ACP prediction. The ACP-DPE consisted of two parallel channels: one was an embedding layer followed by the bi-directional gated recurrent unit (Bi-GRU) module, and the other was an adaptive embedding layer followed by the dilated convolution module. The Bi-GRU module captured the peptide sequence dependencies, whereas the dilated convolution module characterised the local relationship of amino acids. Experimental results show that ACP-DPE achieves an accuracy of 82.81% and a sensitivity of 86.63%, surpassing the state-of-the-art method by 3.86% and 5.1%, respectively. These findings demonstrate the effectiveness of ACP-DPE for ACP prediction and highlight its potential as a valuable tool in cancer treatment research.

## 1 | Introduction

Cancer as a complex and heterogeneous disease that is increasingly becoming the leading cause of death, continues to pose a significant global health burden [1]. According to the cancer report released by the American Cancer Society, 20 million new cancer cases were diagnosed and 9.7 million died from the cancer in 2022 [2]. The number of new cancer cases will rise to 35 million in 2050. Currently, traditional cancer treatment includes surgical therapy, radiotherapy, chemotherapy and targeted therapy. Despite enormous effort paid, no

effective therapy has been developed so far. Radiotherapy or chemotherapy would result in side effects, and chemotherapy would cause drug resistance. Therefore, it is a challenging task to develop a new therapy for cancer treatment.

Anticancer peptides (ACPs), short bioactive peptides usually composed of 5–30 amino acids, have emerged as promising candidates for cancer treatment owing to their high selectivity, high penetration, easy modifications and reduced likelihood of inducing drug resistance [3–8]. Compared to conventional drug therapy, ACPs are of high specificity, low side effects and

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2025 The Author(s). *IET Systems Biology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

acceptable tumour penetration [9]. Most ACPs are derived from natural sources and few are designed synthetically. The discovery and development of ACPs remains challenging due to the vastness of the chemical space and the complexity of their interaction mechanisms with biological targets. Experimental identification methods are often labour intensive, costly, and impractical for large-scale applications [10]. In contrast, computational approaches offer an efficient alternative, enabling high-throughput prediction of ACPs while addressing the limitations of traditional methods. Thus, advancing high-performance ACP prediction models is essential for driving innovation in cancer drug discovery.

Recent advancements in artificial intelligence, particularly in machine learning and deep learning (DL), have become powerful tools in bioinformatics [11–16], significantly accelerating the discovery of ACPs [17]. Several studies have leveraged machine learning algorithms to predict ACPs, with varying degrees of success [6, 18–21]. For example, Manavalan et al. introduced MLACP, which incorporated support vector machine and random forest models, demonstrating superior performance in identifying ACPs [19]. Wei et al. employed a feature representation learning scheme to improve anticancer peptide prediction [20]. These approaches facilitated the identification of ACPs, but their accuracy and efficiency largely depend on the quality and quantity of the training data.

Deep learning as a next-generation artificial intelligence technique, has gained popularity in ACP prediction [8, 22–33]. Lv et al. employed a pre-trained language model and bi-directional long-short-term memory (Bi-LSTM) to learn a deep representation of ACPs [24]. Schaduangrat et al. stacked deep learning to learn spatial and probabilistic feature representations for anticancer peptides [31]. Chen et al. proposed a convolutional neural network (CNN) and multi-task learning-based method for the prediction of ACP activity [23]. Ghulam et al. utilised two-dimensional CNNs (ACP-2DCNN) to enhance the performance of anticancer prediction [25]. Zhu et al. proposed an ACP-check for anticancer peptide prediction which utilised Bi-LSTM to extract time-dependent representation and fused it and amino acid sequence features [28]. Zhu et al. verified the effectiveness of the ACP check on six experimental datasets. Although most methods obtained high performance in predicting ACPs, the explanation is expected to be promoted. To add an explanation to the prediction, Han et al. developed an ACPred-BMF, which integrated Bi-LSTM with self-attention [26]. Recent research studies have focused more on structural innovations. Cui et al. presented a novel deep learning approach that utilised matrix factorization while incorporating length information to predict ACPs [29]. Yuan et al. constructed an ensemble model combining machine learning and deep learning, integrating three channels—CNN, BiLSTM, and LightGBM—by incorporating positional encoding and manual features [33]. Zhong et al. employed a graph attention network (GAT) and LightGBM to predict ACPs and activity types [32]. Wang et al. employed the protein sequence pre-training to identify ACPs and activity types [8]. The advantages of DL-based approaches lie in their ability to capture complex patterns within large datasets, resulting in higher prediction accuracy. However, these methods often demand substantial computational resources, posing significant challenges. In addition, the

black-box nature of DL models makes it difficult to understand the underlying mechanisms behind their predictions.

To address the gap, this study proposed a novel dual-channel deep learning model, ACP-DPE, aimed at improving ACP prediction accuracy. The ACP-DPE employed the adaptive embedding and the embedding as the original encoding. The adaptive embedding integrates both amino acid composition information and positional information. The ACP-DPE utilised the bi-directional gated recurrent unit (Bi-GRU) and dilated convolution to capture long-range dependencies and further refine higher-level representations. Specifically, the dual-channel architecture enabled the Bi-GRU to focus on sequential relationships, while the dilated convolution extended the receptive field, facilitating more effective feature sensing. Extensive experiments were conducted to assess the model's robustness and rationality, demonstrating that ACP-DPE outperforms existing methods in terms of both prediction accuracy and generalisation ability.

## 2 | Materials and Methods

### 2.1 | Datasets

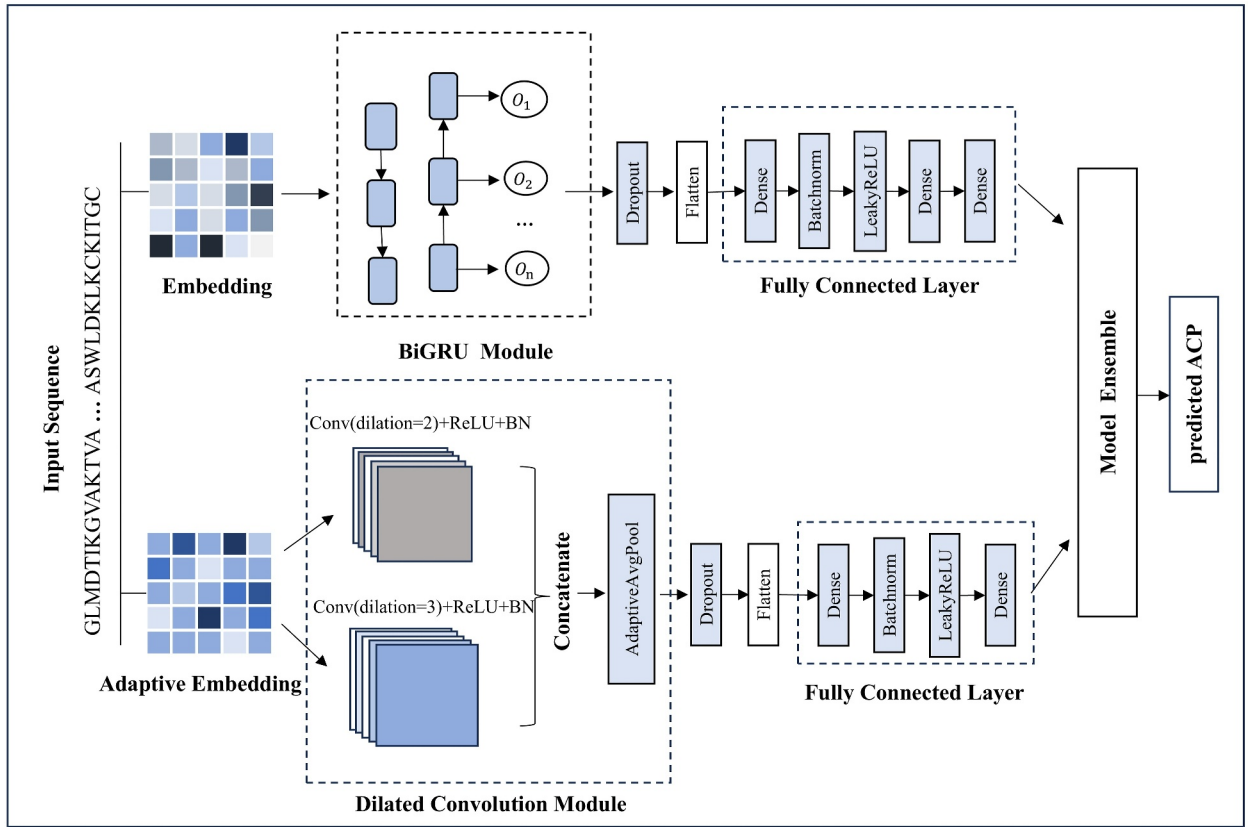
For a fair comparison with the existing method, we employed the same datasets as AntiCP 2.0 [34]. The AntiCP 2.0 provided two datasets: the main dataset and the alternate dataset. The experimentally validated ACPs (positive samples) in both datasets were derived from the CancerPPD database [35], which is a repository of anticancer peptides and proteins. The same number of antimicrobial peptides as the positive samples were used as the negative samples in the main dataset. The negative samples in the alternate dataset are random peptides. The main dataset consisted of 1378 samples which are used both to optimise the parameters and for training, and an additional 344 samples for independent tests. The alternate dataset comprises 1552 samples for training and 388 samples for testing, as shown in Table 1. The rate of the numbers of the positive to the negative samples in both datasets was 1:1, and the maximum length of peptide sequences in the dataset was 50. For samples with less than 50 amino acid residues, the corresponding number of zeros was attached to the end during encoding, which ensured that the sequence had a uniform length. Both datasets are available for download from <https://webs.iitd.edu.in/raghava/anticp2/>.

### 2.2 | Methodology

As shown in Figure 1, the proposed ACP-DPE consisted of dual parallel channels. In the first channel, the embedding layer was followed by the Bi-GRU, which was followed by three fully

**TABLE 1** | Number of the positive and the negative samples used for training and testing.

Dataset	Main dataset		Alternate dataset	
	Positive	Negative	Positive	Negative
Training	689	689	776	776
Testing	172	172	194	194



**FIGURE 1** | Overview of the ACP-DPE. The outputs  $O_1, O_2, \dots, O_n$  are processed values at each timestep from the BiGRU module.

connected layers. The embedding layer was intended to transform the character sequence into a low-dimensional vector representation, and Bi-GRU was utilised to capture dependency relationships within the sequences. In the second channel, adaptive embedding was followed by the dilated convolution module and then by two fully connected layers. The adaptive embedding was used to obtain token and positional information, and dilated convolution was adopted to extract high-level features of the sequences. The outputs of the two channels are integrated for a final decision.

### 1. Embedding and adaptive embedding

In the ACP-DPE, the embedding and the adaptive embedding were used, respectively, in the first layer of each channel. The embedding layer was to convert a sequence of characters into a sequence of integers and then embed the integer sequence into continuous vectors. To obtain a comprehensive representation of peptide sequences, we also adopted adaptive embedding [36]. Adaptive embedding not only characterises the relationships between tokens but also fuses the positional information. Therefore, the adaptive embedding was more flexible and stronger in the representation than the traditional embedding.

The adaptive embedding was described as follows:

$$E_{ad} = E_{tok} + E_{pos} \quad (1)$$

where  $E_{tok}$  represented the token vector embedding, and  $E_{pos}$  represented the positional embedding.

### 2. Bi-GRU

In the proposed ACP-DPE, Bi-GRU was utilised to refine semantics from the embedding layer and characterise the contextual relationship between tokens. The GRU [37] was of simpler structure and had fewer gating units than the LSTM [38]. The GRU was equipped with an updated gate  $z_t$  and a reset gate  $\gamma_t$ .

The GRU was computed as follows:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (2)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (3)$$

where  $h_t$  was the hidden state at time  $t$ ,  $\tilde{h}_t$  was the candidate's hidden state,  $x_t$  was the input at time  $t$ ,  $h_{t-1}$  was the hidden state of the layer at time  $t - 1$  or the initial hidden state at time 0, and  $W, U, b$  represented two sets of weights and a bias term.  $\odot$  denoted the Hadamard product.

$z_t$  and  $\gamma_t$  were, respectively, computed using the following equation:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (4)$$

$$\gamma_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (5)$$

where  $\sigma$  was the *sigmoid* activation function. A dropout layer with a rate of 0.5 was attached to the Bi-GRU to reduce overfitting.

### 3. Dilated convolution

The ACP-DPE used the dilated convolution to capture features. Dilated convolution is a special type of convolution where the kernel is applied with gaps between the elements, thereby increasing the receptive field without increasing the computational cost or the number of parameters [39].

The dilated convolution can be described as follows [40]:

$$y_i = \sum_k x_{i+d \cdot k} w_k \quad (6)$$

where  $y_i$  represented the  $i$ th value of the output sequence,  $x$  represented the input sequence,  $w_k$  was associated with the  $k \times k$  convolution kernel, and  $d$  is the dilated ratio.

This module used two dilated convolution layers (conv1 and conv2), with each convolution layer focussing on feature extraction and expanding the receptive field. The dilated ratios were 2 and 3, and the kernel size was 4.

Each convolutional layer shared the same structure, including convolution operation, ReLU activation function, and batch normalisation.

The operation of the convolution layer was described as follows:

$$C = \text{BN}(\text{ReLU}(\text{Conv}(E_{\text{ad}}))) \quad (7)$$

where  $E_{\text{ad}}$  represented the outputs of the adaptive embedding. Conv was the convolution operation. Subsequently, the outputs of the two convolutional layers were concatenated in parallel, followed by the application of adaptive average pooling (AdaptiveAvgPool) to dynamically adjust the output to a predefined size. This process reduced the dimensionality of the fused multi-scale high-level features.

The AdaptiveAvgPool operation was described as follows:

$$P = \text{AdaptiveAvgPool}(\text{Concate}(C_1, C_2)) \quad (8)$$

where  $C_1$  and  $C_2$  were the outputs of the two convolutional layers. To reduce over-fitting, the dropout layer with a rate of 0.4 was attached to the AdaptiveAvgPool layer.

### 4. Fully Connected Layer

In each channel, the fully connected layer was intended to map the multi-dimensional feature representations to the target output dimension. The first linear layer after each channel applied batch normalisation and LeakyReLU activation function. The batch normalisation was used to accelerate convergence and the LeakyReLU activation was used to stabilise the gradients during training.

The outputs of the dual channels were computed as follows:

$$\text{output1} = \sigma(f_1(f_2(\text{BN}(f_1(\text{Bo}_1)))))) \quad (9)$$

$$\text{output2} = \sigma(f_1(f_2(\text{BN}(f_1(\text{Do}_2)))))) \quad (10)$$

where  $\sigma$  was the sigmoid activation function,  $f_1$  was the dense operation, and  $f_2$  was the LeakyReLU activation function.  $\text{Bo}_1$  represented the flattened output from the Bi-GRU channel, whereas  $\text{Do}_2$  corresponded to the flattened output from the dilated convolution channel. Dense was the fully connected layer.

The ACP-DPE integrated the outputs from both channels to make the final decision, which was computed using the following equation:

$$\text{output} = a_1 * \text{output1} + a_2 * \text{output2} \quad (11)$$

where  $a_1$  and  $a_2$  were the predefined weight.

### 5. Loss function

We employed binary cross-entropy (BCELoss) as the loss function, which was defined as follows [41]:

$$L(y, p) = -(y \log p + (1 - y) \log(1 - p)) + \lambda \|W\|^2 \quad (12)$$

where  $y$  represented the true labels, and  $p$  represented the predicted values.  $\lambda \|W\|^2$  was the regularisation term, where  $\lambda$  was weight decay [42], and  $\|W\|^2$  was the weight parameter of the model.

We used the Adam optimiser [42] to update the learning parameters. The weight decay is set to  $9e - 03$ . Additionally, the learning rate is set to 0.001, the number of epochs to 100, and the batch size to 128.

### 6. Evaluation metrics

Five metrics were used to measure the performances, which were, respectively, sensitivity (SN), specificity (SP), accuracy (ACC), Matthew correlation coefficient (MCC) [43], and the area under the curve (AUC). They were defined as follows:

$$\text{SN} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (13)$$

$$\text{SP} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (14)$$

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (15)$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (16)$$

$$\text{AUC} = \frac{\sum_{i \in \text{positive class}} \text{rank}_i - \frac{M \times (1 + M)}{2}}{M \times N} \quad (17)$$

Here, TP represented true positive, TN denoted true negative, FP represented false positive, and FN was false negative.  $M$  and  $N$  represented the number of positive and negative samples, respectively. AUC was the area under the receiver operating characteristic (ROC) curve which was linking the true positive rates against the false positive rates under various thresholds [44]. The AUC reflects the entire performance under different thresholds. The AUC ranges from 0 to 1. One means the perfect

performance, 0.5 is the random guess, and 0 is the reverse prediction. The more the AUC, the better the performance.

### 3 | Results

#### 3.1 | Comparison With Different Encoding

To test the embedding and adaptive embedding for effectiveness, we compared them with traditional encoding methods, including BLOSUM62, AAindex, ZScale, and binary encoding. BLOSUM62 reflected the evolutionary characteristics of amino acid residues, AAindex and ZScale described the physico-chemical properties, and binary encoding was the composition of amino acid. For the powerful ability to represent biological sequences, the four encodings above were widely applied to protein sequence analysis such as malonylation site prediction [45], ubiquitylation site prediction [46] and sumoylation site prediction [47]. For a fair comparison, the ACP-DPE replaced one encoding in one branch each time while keeping the other unchanged.

As illustrated in Figure 2a, the embedding method performed exceptionally well in the Bi-GRU branch, achieving the highest values for ACC (0.8047) and MCC (0.6109). Compared to the AAindex method, although the SP and AUC of the embedding method decreased slightly by 0.75% and 0.41%, respectively, it significantly improved SN (3.26%), ACC (1.17%) and MCC (2.51%). Although the ZScale method achieved the highest SN, its performance on other metrics was suboptimal. The adaptive embedding, Binary, and BLOSUM62 methods demonstrate relatively balanced performance across various metrics, though their overall performance was slightly inferior to those of the embedding method.

As shown in Figure 2b, in the dilated convolution branch, the adaptive embedding method demonstrated superior performance, achieving the highest values for SN (0.8293), ACC (0.8047), MCC (0.6109) and AUC (0.8538). Compared to the embedding method, although the SP of the adaptive embedding method decreased by 4.51%, it significantly improved SN by

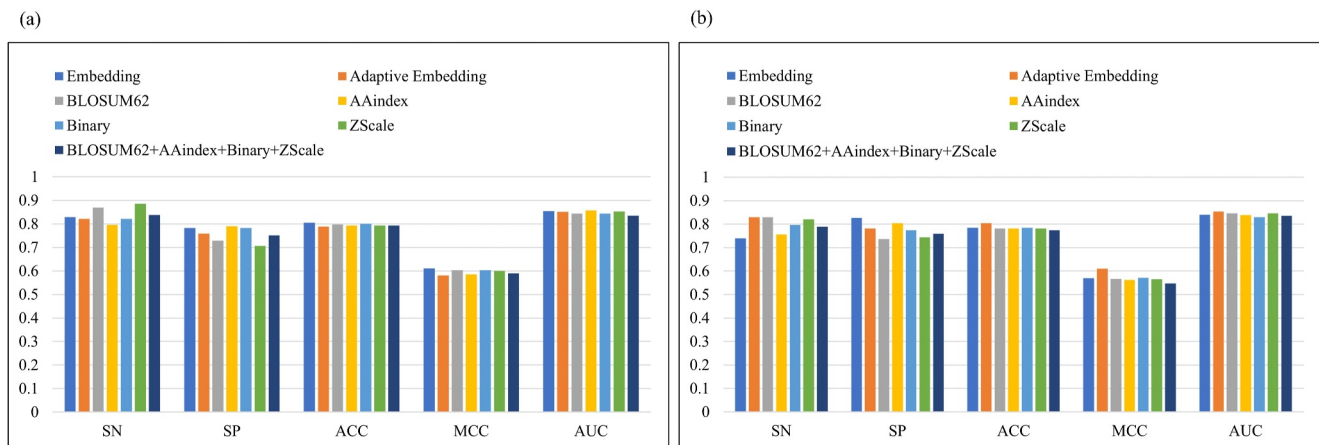
8.95%, ACC by 1.95%, MCC by 4.10% and AUC by 1.43%. The performance of the four individual manual feature encodings was relatively poor, with the combined performance being the lowest. These results suggest that the proposed ACP-DPE framework is effective in predicting anticancer peptides by utilising the embedding method in the Bi-GRU branch and the adaptive embedding method in the dilated convolution branch.

#### 3.2 | Model Selection and Parameters Optimization

To validate the rationality of the Bi-GRU branch in the ACP-DPE, we conducted a comparison with three other deep learning modules (i.e., Bi-LSTM, Bi-GRU + MultiAttention and Bi-LSTM + MultiAttention). Table 2 shows the performances of four methods. From Table 2, it is evident that Bi-GRU outperformed Bi-LSTM, the combination of Bi-GRU and MultiAttention and the combination of Bi-LSTM and MultiAttention. Specifically, Bi-GRU achieved a 3.01% increase in SP, a 1.17% increase in ACC and a 0.0209 increase in MCC compared to Bi-LSTM. Bi-GRU also improved SN by 13.82%, ACC by 1.17%, MCC by 0.0187 and AUC by 1.07% over the combination of Bi-GRU and MultiAttention. Furthermore, Bi-GRU enhanced SN by 6.51%, ACC by 1.56%, MCC by 3.37%, and AUC by 0.03% compared to the combination of Bi-LSTM and MultiAttention. The results demonstrated the rationality of using Bi-GRU in the ACP-DPE.

Table 3 lists the performance of Bi-GRU with different hidden sizes in ACP-DPE. The hidden size varies between 8 and 32, rising at an equal ratio of 2. The Bi-GRU with the 16 achieves the highest ACC of 80.47%. Therefore, we set the hidden size of Bi-GRU to 16.

Similarly, we compared multiple CNNs. Namely, we compared a single convolutional layer, two convolutional layers in parallel and three convolutional layers in parallel. Each convolutional layer had the same structure, consisting of a convolution operation, ReLU activation function and batch normalisation. Table 4 shows the performance of different CNN architectures in the dilated convolution module.



**FIGURE 2** | Performance comparison of different encodings in the ACP-DPE model: (a) varying the encoding only in the Bi-GRU channel and (b) varying the encoding only in the dilated convolution channel. Each experiment tests the impact of different features on the model's performance by changing one channel's encoding while keeping the other fixed.



**TABLE 2** | Performance comparison of different modules in the Bi-GRU branch.

Model	SN	SP	ACC	MCC	AUC
Bi-LSTM	<b>83.74</b>	75.19	79.30	0.5900	<b>85.62</b>
Bi-GRU	82.93	78.20	<b>80.47</b>	<b>0.6109</b>	85.38
Bi-GRU + MultiAttention	69.11	<b>88.72</b>	79.30	0.5922	84.31
Bi-LSTM + MultiAttention	76.42	81.20	78.91	0.5772	85.35

Note: Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

**TABLE 3** | Performance comparison of Bi-GRU with different hidden sizes in ACP-DPE.

Hidden size	SN	SP	ACC	MCC	AUC
8	<b>85.37</b>	75.19	80.08	0.6069	<b>86.58</b>
16	82.93	<b>78.20</b>	<b>80.47</b>	<b>0.6109</b>	85.38
32	80.49	77.44	78.91	0.5789	84.98

Note: Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

**TABLE 4** | Performance comparison of different CNN layers in dilated convolution.

Layers	SN	SP	ACC	MCC	AUC
1	82.93	76.69	79.69	0.5962	84.39
2	<b>82.93</b>	<b>78.20</b>	<b>80.47</b>	<b>0.6109</b>	<b>85.38</b>
3	81.30	78.20	79.69	0.5945	84.88

Note: Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

The dilated convolution with two convolutional layers in parallel achieved the highest SN, ACC, MCC, and AUC, surpassing the one with three convolutional layers in parallel by 1.63%, 0.78%, 1.64% and 0.5%, respectively. A single convolutional layer and three convolutional layers in parallel reached the same ACC, which meant no significant difference in overall performance. Therefore, we employ dilated convolution with two convolutional layers in parallel. We experimentally tested hyper-parameters in the dilated convolution module on the main dataset. Table 5 presents the performances of different dilated radio combinations, Table 6 displays the performance of different convolutional kernel size combinations, and Table 7 provides a comparison of different output channels in the dilated convolution module. A dilated ratio combination of {2, 3}, a kernel size combination of {4, 4} and an output channel value of 256 performed better. Therefore, we used these parameters to configure the ACP-DPE.

### 3.3 | Comparison With Existing Methods

To validate the efficiency of the proposed ACP-DPE, we compared it with the existing methods, namely, ACP-OPE (the main dataset is missing four samples) [33], iACP-DRLF (the main dataset and alternate dataset are both missing four samples) [24], AntiCP\_2.0 [34], ACPred-Fuse [48], iACP [6], ACPred [31], PEPred-Suite [49], AntiCP [50], and ACPred-FL [20]. As shown in Table 8, on the main dataset, the proposed ACP-DPE obtained the SN of 86.63%, and the ACC of 82.81%,

**TABLE 5** | Performance comparison of different dilated ratio combinations in dilated convolution.

Dilated radio	SN	SP	ACC	MCC	AUC
{1, 2}	77.24	<b>81.20</b>	79.30	0.5851	84.31
{1, 3}	81.30	77.44	79.30	0.5871	<b>85.71</b>
{2, 3}	<b>82.93</b>	78.20	<b>80.47</b>	<b>0.6109</b>	85.38

Note: Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

**TABLE 6** | Performance comparison of different convolutional kernel size combinations in dilated convolution.

Kernel size	SN	SP	ACC	MCC	AUC
{2, 2}	76.42	<b>81.20</b>	78.91	0.5772	84.67
{3, 3}	80.49	79.70	80.08	0.6015	<b>86.12</b>
{4, 4}	82.93	78.20	<b>80.47</b>	<b>0.6109</b>	85.38
{5, 5}	<b>83.74</b>	76.69	80.08	0.6045	84.18

Note: Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

**TABLE 7** | Performance comparison of different output channels in dilated convolution.

Output channel	SN	SP	ACC	MCC	AUC
16	<b>86.18</b>	72.93	79.30	0.5942	<b>86.39</b>
32	82.11	76.69	79.30	0.5879	85.38
64	83.74	76.69	80.08	0.6045	85.12
128	83.74	75.19	79.30	0.5900	85.30
256	82.93	78.20	<b>80.47</b>	<b>0.6109</b>	85.38
512	81.30	<b>79.70</b>	80.47	0.6096	84.54

Note: Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

elevating SN by 5.1%, and ACC by 3.86% over the ACP-OPE, respectively. On the alternate dataset, the proposed ACP-DPE reached almost equal performance with the iACP-DRLF.

### 3.4 | Ablation Study

To further demonstrate the effectiveness of the ACP-DPE architecture, we conducted ablation experiments on both the main dataset and the alternate dataset. As shown in Table 9, on the main dataset, the ACP-DPE reached a SN of 86.63%, an ACC of 82.81%, a MCC of 0.6126 and an AUC of 83.62%, outperforming the other five architectures. For example, compared to the dual-channel with Bi-GRU, ACP-DPE improved SN by

**TABLE 8** | Performance comparison with existing methods.

Method	Main dataset				Alternate dataset			
	SN	SP	ACC	MCC	SN	SP	ACC	MCC
ACP-DPE	<b>86.63</b>	75.00	<b>82.81</b>	<b>0.61</b>	<b>92.27</b>	93.68	92.97	<b>0.86</b>
ACP-OPE <sup>b</sup>	81.53	<b>76.76</b>	78.95	—	—	—	—	—
iACP-DRLF <sup>b</sup>	80.7	74.3	77.5	0.55	89.6	<b>96.4</b>	<b>93.0</b>	0.86
AntiCP_2.0 <sup>a</sup>	77.46	73.41	75.43	0.51	92.27	91.75	92.01	0.84
AntiCP <sup>a</sup>	100.00	1.16	50.58	0.07	89.69	90.20	89.95	0.80
ACPred <sup>a</sup>	85.55	21.39	53.47	0.09	87.11	83.51	85.31	0.71
ACPred-FL <sup>a</sup>	67.05	22.54	44.80	−0.12	60.21	25.58	43.80	−0.15
ACPred-fuse <sup>a</sup>	69.19	68.60	68.90	0.38	64.43	93.30	78.87	0.60
PEPred-suite <sup>a</sup>	33.14	73.84	53.49	0.08	40.21	74.74	57.47	0.16
iACP <sup>a</sup>	77.91	32.16	55.10	0.11	78.35	76.80	77.58	0.55

Note: Bold values indicate the highest value in each metric.

<sup>a</sup>Results sourced from AntiCP\_2.0 [34].

<sup>b</sup>Results sourced from the respective method [24, 33]. Units: percentages (%) for SN, SP, ACC.

**TABLE 9** | Performance comparison of Bi-GRU, dilated convolution, and their different combinations in the dual-channel on the main dataset.

Model	SN	SP	ACC	MCC	AUC
Baseline (without)	78.49	69.05	75.39	0.4620	78.74
Bi-GRU (single)	77.33	<b>76.19</b>	76.95	0.5124	82.52
Dilated convolution (single)	77.33	73.81	76.17	0.4912	82.09
Bi-GRU (dual)	82.56	73.81	79.69	0.5516	83.53
Dilated convolution (dual)	86.05	58.33	76.95	0.4615	81.17
Bi-GRU + dilated convolution	<b>86.63</b>	75.00	<b>82.81</b>	<b>0.6126</b>	<b>83.62</b>

Note: Baseline (without) refers to the ACP-DPE without Bi-GRU and dilated convolution. ‘Single’ denotes a model with only one channel, while ‘dual’ refers to both channels. Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

4.07%, SP by 1.19%, ACC by 3.12%, MCC by 6.1% and AUC by 0.09%. In comparison with the single-channel with Bi-GRU, although the SP was slightly lower by 1.19%, the ACP-DPE significantly promoted ACC by 5.86%. As shown in Table 10, on the alternate dataset, both the dual-channel with dilated convolution and ACP-DPE achieved the highest ACC. Additionally, the ACP-DPE improved SN by 0.52%, and AUC by 0.17%. Therefore, the ACP-DPE showed overall advantages over other single or dual-channel architectures.

### 3.5 | Robustness

To test the robustness of ACP-DPE, we introduced a certain proportion of noise to positive samples in the training set of the main dataset. Specifically, we randomly selected a proportion of positive samples from the training set and randomly exchanged 20% of the amino acid residues for each selected sample. The exchanged samples were considered positive samples, which were used for training the ACP-DPE along with no changed samples. Table 11 shows the performance of the ACP-DPE under various ratios of the exchanged to the total positive samples. All evaluation metrics showed a decrease when the ratio of noise to positive samples increased. When the noise ratio reached 1%, SN decreased by 2.91%, SP by 4.76%, ACC by 3.51%, MCC by 7.77% and AUC by 1.35% compared with the initial state (no noise). This suggests that the inclusion of noise affects the performance of the

ACP-DPE. When the ratio rises from 2% to 5%, the ACCs and the MCCs begin to significantly reduce, and then tend to be stable and reduce remarkably at the ratio of 5%. The AUCs seem to change little. These results indicated that the ACP-DPE exhibits a certain degree of robustness against such disturbances.

### 3.6 | Visualisation Analysis

To show the ability of learning representations to differentiate ACP from non-ACP, we visualise the output in the various layers by using the UMAP (uniform manifold approximation and projection) [51]. The UMAP is a manifold learning-based method for dimension reduction. Due to its outstanding ability to reduce dimensions, the UMAP has widely been applied to visualise biological molecular high-dimensional data [52, 53]. As shown in Figures 3a–d, the ACPs and the non-ACPs became remarkably distinguished through the dilated convolution and the Bi-GRU channel, indicating that the deep neural network learns well the representation of the ACP.

### 3.7 | Web Server

We have deployed the ACP-DPE on the webserver for free use. The homepage includes introductions to anticancer

**TABLE 10** | Performance comparison of Bi-GRU, dilated convolution, and their different combinations in the dual-channel on the alternate dataset.

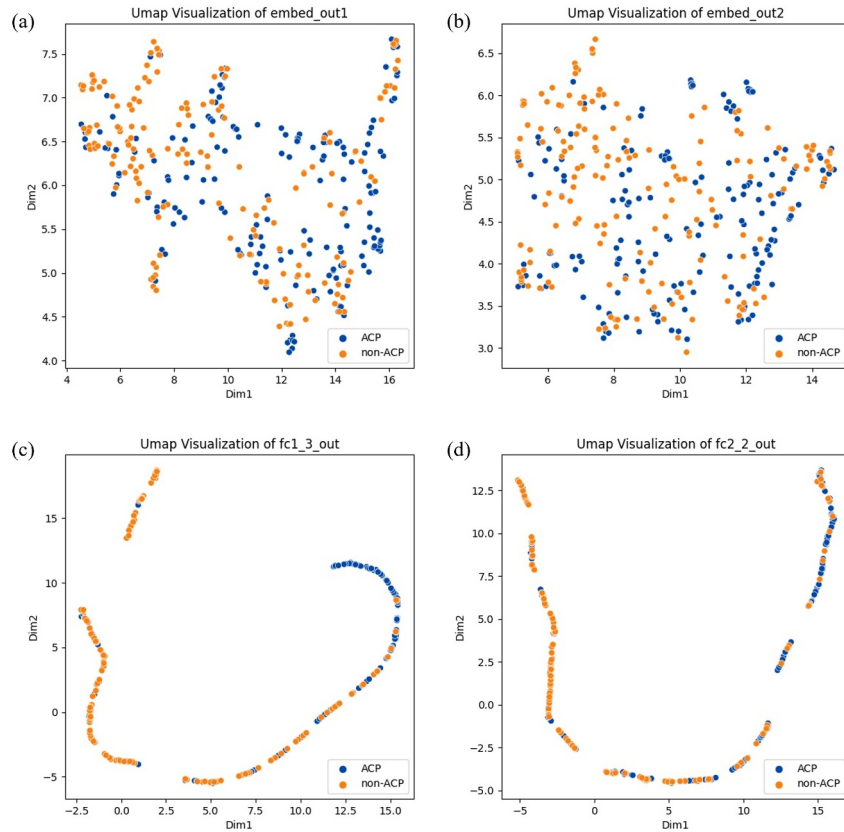
Model	SN	SP	ACC	MCC	AUC
Baseline (without)	86.60	92.11	89.32	0.7879	93.36
Bi-GRU (single)	91.24	91.05	91.15	0.8229	96.60
Dilated convolution (single)	91.24	93.16	92.19	0.8440	95.99
Bi-GRU (dual)	90.72	92.11	91.41	0.8282	96.37
Dilated convolution (dual)	91.75	<b>94.21</b>	<b>92.97</b>	<b>0.8597</b>	97.20
Bi-GRU + dilated convolution	<b>92.27</b>	93.68	92.97	0.8595	<b>97.37</b>

Note: Baseline (without) refers to the ACP-DPE without Bi-GRU and dilated convolution. ‘Single’ denotes a model with only one channel, while ‘dual’ refers to both channels. Units: percentages (%) for SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.

**TABLE 11** | Performance comparison after shuffling amino acid residues at different ratios of positive samples.

Ratio	SN	SP	ACC	MCC	AUC
0	<b>86.63</b>	75.00	<b>82.81</b>	<b>0.6126</b>	83.62
1	83.72	70.24	79.30	0.5349	82.27
2	81.40	76.19	79.69	0.5585	<b>84.94</b>
3	80.23	<b>78.57</b>	79.69	0.5659	83.94
4	84.88	71.43	80.47	0.5598	82.92
5	79.65	73.81	77.73	0.5174	83.90

Note: Units: percentages (%) for ratio, SN, SP, ACC, and AUC. Bold values indicate the highest value in each metric.



**FIGURE 3** | Visualisation of model-specific layer outputs using UMAP dimensionality reduction. (a) The embedding layer in the Bi-GRU branch, (b) the embedding layer in the dilated convolution branch, (c) the last fully connected layer in the Bi-GRU branch and (d) the last fully connected layer in the dilated convolution branch.



peptides and the ACP-DPE. Users navigate to the ‘WEB-SERVER’ interface (Figure 4) to upload the sequence data. Users have two options to upload sequence data. The first is to paste the fasta sequence into the textbox. The second is to upload files of sequences in the fasta format. Users can click here to obtain some examples of fasta sequences. The web server required all the sequences not to exceed 50 amino acid residues. By clicking the submit button, users can obtain the predicted results. The time when users wait depends not only on the number of uploaded sequences but also on the internet network speed. In addition, the webserver also provides the experimental datasets for users to download.

#### 4 | Discussion

We proposed the ACP-DPE method for identifying anticancer peptides, consisting of dual parallel channels. The first channel employs an embedding layer to transform the character sequence into a low-dimensional vector representation, followed by Bi-GRU to capture dependency relationships within the sequences. The second channel employs adaptive embedding to characterise both token and positional information, followed by dilated convolution to extract high-level sequence features. ACP-DPE significantly improved performance through dual-channel heterogeneous fusion. However, its effectiveness depends on the quality and quantity of the training data. When the sample size is too small, the model may not fully learn essential features. As shown in Tables 9 and 10, ACP-DPE with dual channels performed better than a single channel, likely because dual channels provide more comprehensive representations. In Table 9, the dual heterogeneous channels (Bi-GRU + dilated convolution) outperformed the dual homogeneous channels, as the heterogeneous channels capture complementary features.

In contrast, the inclusion of multi-attention mechanisms did not improve the ACP-DPE’s performance (Table 2). Although attention mechanisms, which dynamically adjust feature weights, are widely used in models such as transformers to enhance performance, their effectiveness may be limited by

insufficient training data. State-of-the-art methods for ACP prediction, such as ACP-OPE [33] and iACP-DRLF [24], adopt different strategies. ACP-OPE integrated deep learning and traditional feature representation, whereas iACP-DRLF fused two pre-trained embeddings. In comparison, our ACP-DPE employs a dual-channel architecture (Bi-GRU and dilated convolution) to fuse heterogeneous representations. As shown in Table 8, ACP-DPE achieved a 3.86% improvement in **ACC** over ACP-OPE on the main dataset. Additionally, it outperformed iACP-DRLF by 2.67% in **SN** on the alternative dataset. This improvement would be attributed to the use of adaptive embedding and embedding within the dual-channel heterogeneous architecture, allowing the model to dynamically adjust representations according to the task requirements.

#### 5 | Conclusion

Anticancer peptides play a key role in cancer treatment, but the discovery of ACPs is still challenging. In this study, we proposed ACP-DPE, a dual-channel-based deep learning method to enhance ACP prediction. ACP-DPE integrates both adaptive embedding and standard embedding, dynamically adjusting feature extraction based on the task requirements. The model utilises heterogeneous channels to refine high-level representation. One channel uses Bi-GRU to capture the temporal dependence within sequences, whereas the other uses dilated convolutions to extract local features. This dual-channel architecture effectively fuses different feature representations. Experimental results demonstrate that ACP-DPE outperforms existing methods, achieving higher prediction accuracy and robustness. These findings indicate that ACP-DPE has the potential to be a valuable tool in ACP identification and cancer treatment research.

However, ACP-DPE faces challenges with sequences of varying lengths, especially with longer sequences. In future work, we will focus on enhancing ACP prediction for large datasets with variable sequence lengths, potentially improving the robustness and generalisability of ACP-DPE across diverse sequence datasets.

ACP-DPE: a dual-channel deep learning model for anticancer peptide prediction

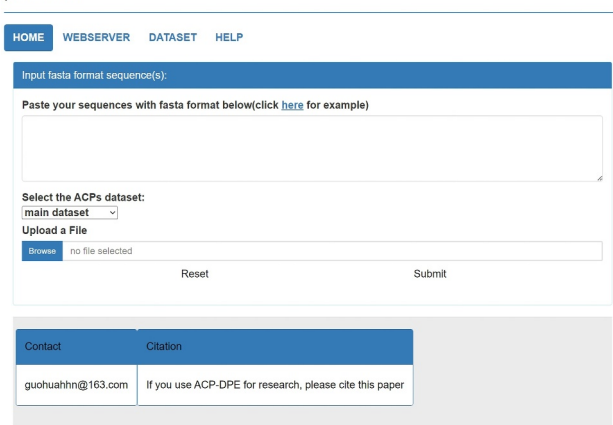


FIGURE 4 | ACP-DPE web server interface.

#### Author Contributions

**Guohua Huang:** conceptualisation, methodology, supervision, writing. **Yujie Cao:** data curation, writing – original preparation, software, validation. **Qi Dai:** conceptualisation, supervision. **Weihong Chen:** supervision, validation, writing – review and editing. All the authors read and approved the manuscript.

#### Ethics Statement

The authors have nothing to report.

#### Consent

The authors have nothing to report.

#### Conflicts of Interest

The authors declare no conflicts of interest.

## Data Availability Statement

All the experimental data were available at <https://github.com/CYJ-sudo/ACP-DPE/tree/main>.

## References

1. J. Ferlay, M. Ervik, F. Lam, et al., *Global Cancer Observatory: Cancer Today* (International Agency for Research on Cancer, 2020): Cancer Tomorrow. (2021).
2. American Cancer Society, American Cancer Society Releases Latest Global Cancer Statistics; Cancer Cases Expected to Rise to 35 Million Worldwide by 2050, (2024), <https://pressroom.cancer.org/GlobalCancerStatistics2024>.
3. L. Otvos, "Peptide-based Drug Design: Here and Now," *Peptide-Based Drug Design* 494 (2008): 1–8, [https://doi.org/10.1007/978-1-59745-419-3\\_1](https://doi.org/10.1007/978-1-59745-419-3_1).
4. P. Vlieghe, V. Lisowski, J. Martinez, and M. Khrestchatsky, "Synthetic Therapeutic Peptides: Science and Market," *Drug Discovery Today* 15, no. 1–2 (2010): 40–56, <https://doi.org/10.1016/j.drudis.2009.10.009>.
5. J. Thundimadathil, "Cancer Treatment Using Peptides: Current Therapies and Future Prospects," *Journal of Amino Acids* 2012, no. 1 (2012): 1–13, <https://doi.org/10.1155/2012/967347>.
6. W. Chen, H. Ding, P. Feng, H. Lin, and K.-C. Chou, "iACP: A Sequence-Based Tool for Identifying Anticancer Peptides," *Oncotarget* 7, no. 13 (2016): 16895–16909, <https://doi.org/10.18632/oncotarget.7815>.
7. W. Chiangjong, S. Chutipongtanate, and S. Hongeng, "Anticancer Peptide: Physicochemical Property, Functional Aspect and Trend in Clinical Application," *International Journal of Oncology* 57, no. 3 (2020): 678–696, <https://doi.org/10.3892/ijo.2020.5099>.
8. S. Wang and B. Ma, "Anti-cancer Peptides Identification and Activity Type Classification With Protein Sequence Pre-Training," *IEEE Journal of Biomedical and Health Informatics* (2024): 1–10, <https://doi.org/10.1109/JBHI.2024.3358632>.
9. P. Norouzi, M. Mirmohammadi, and M. H. H. Tehrani, "Anticancer Peptides Mechanisms, Simple and Complex," *Chemico-Biological Interactions* 368 (2022): 110194, <https://doi.org/10.1016/j.cbi.2022.110194>.
10. A. Gautam, K. Chaudhary, R. Kumar, et al., "In Silico Approaches for Designing Highly Effective Cell Penetrating Peptides," *Journal of Translational Medicine* 11 (2013): 1–12, <https://doi.org/10.1186/1479-5876-11-74>.
11. H. Wu, Y. Wu, Y. Jiang, et al., "scHiCStackL: A Stacking Ensemble Learning-Based Method for Single-Cell Hi-C Classification Using Cell Embedding," *Briefings in Bioinformatics* 23, no. 1 (2022), <https://doi.org/10.1093/bib/bbab396>.
12. X. Yang, K. K. Mann, H. Wu, and J. Ding, "scCross: A Deep Generative Model for Unifying Single-Cell Multi-Omics With Seamless Integration, Cross-Modal Generation, and In Silico Exploration," *Genome Biology* 25, no. 1 (2024): 198, <https://doi.org/10.1186/s13059-024-03338-z>.
13. Y. Wu, Z. Shi, X. Zhou, et al., "scHiCyclePred: A Deep Learning Framework for Predicting Cell Cycle Phases From Single-Cell Hi-C Data Using Multi-Scale Interaction Information," *Communications Biology* 7, no. 1 (2024): 923, <https://doi.org/10.1038/s42003-024-06626-3>.
14. B. Zhou, Q. Liu, M. Wang, and H. Wu, "Deep Neural Network Models for Cell Type Prediction Based on Single-Cell Hi-C Data," *BMC Genomics* 22, no. Suppl 5 (2021): 922, <https://doi.org/10.1186/s12864-024-10764-7>.
15. P. Zhang, Y. Wu, H. Zhou, B. Zhou, H. Zhang, and H. Wu, "CLNN-Loop: A Deep Learning Model to Predict CTCF-Mediated Chromatin Loops in the Different Cell Lines and CTCF-Binding Sites (CBS) Pair Types," *Bioinformatics* 38, no. 19 (2022): 4497–4504, <https://doi.org/10.1093/bioinformatics/btac575>.
16. Z. Shi and H. Wu, "CTPredicator: A Comprehensive and Robust Framework for Predicting Cell Types by Integrating Multi-Scale Features From Single-Cell Hi-C Data," *Computers in Biology and Medicine* 173 (2024): 108336, <https://doi.org/10.1016/j.combiomed.2024.108336>.
17. X. Liang, F. Li, J. Chen, et al., "Large-Scale Comparative Review and Assessment of Computational Methods for Anti-Cancer Peptide Identification," *Briefings in Bioinformatics* 22, no. 4 (2021), <https://doi.org/10.1093/bib/bbaa312>.
18. Z. Hajisharifi, M. Piryaiee, M. M. Beigi, M. Behbahani, and H. Mohabatkari, "Predicting Anticancer Peptides With Chou's Pseudo Amino Acid Composition and Investigating Their Mutagenicity Via Ames Test," *Journal of Theoretical Biology* 341 (2014): 34–40, <https://doi.org/10.1016/j.jtbi.2013.08.037>.
19. B. Manavalan, S. Basith, T. H. Shin, S. Choi, M. O. Kim, and G. Lee, "MLACP: Machine-Learning-Based Prediction of Anticancer Peptides," *Oncotarget* 8, no. 44 (2017): 77121–77136, <https://doi.org/10.18632/oncotarget.20365>.
20. L. Wei, C. Zhou, H. Chen, J. Song, and R. Su, "ACPred-FL: A Sequence-Based Predictor Using Effective Feature Representation to Improve the Prediction of Anti-Cancer Peptides," *Bioinformatics* 34, no. 23 (2018): 4007–4016, <https://doi.org/10.1093/bioinformatics/bty451>.
21. V. Boopathi, S. Subramaniam, A. Malik, G. Lee, B. Manavalan, and D. C. Yang, "mACPPred: A Support Vector Machine-Based Meta-Predictor for Identification of Anticancer Peptides," *International Journal of Molecular Sciences* 20, no. 8 (2019): 1964, <https://doi.org/10.3390/ijms20081964>.
22. L. Yu, R. Jing, F. Liu, J. Luo, and Y. Li, "DeepACP: A Novel Computational Approach for Accurate Identification of Anticancer Peptides by Deep Learning Algorithm," *Molecular Therapy - Nucleic Acids* 22 (2020): 862–870, <https://doi.org/10.1016/j.omtn.2020.10.005>.
23. J. Chen, H. H. Cheong, and S. W. I. Siu, "xDeep-AcPEP: Deep Learning Method for Anticancer Peptide Activity Prediction Based on Convolutional Neural Network and Multitask Learning," *Journal of Chemical Information and Modeling* 61, no. 8 (2021): 3789–3803, <https://doi.org/10.1021/acs.jcim.1c00181>.
24. Z. Lv, F. Cui, Q. Zou, L. Zhang, and L. Xu, "Anticancer Peptides Prediction With Deep Representation Learning Features," *Briefings in Bioinformatics* 22, no. 5 (2021), <https://doi.org/10.1093/bib/bbab008>.
25. A. Ghulam, F. Ali, R. Sikander, A. Ahmad, A. Ahmed, and S. Patil, "ACP-2DCNN: Deep Learning-Based Model for Improving Prediction of Anticancer Peptides Using Two-Dimensional Convolutional Neural Network," *Chemometrics and Intelligent Laboratory Systems* 226 (2022): 104589, <https://doi.org/10.1016/j.chemolab.2022.104589>.
26. B. Han, N. Zhao, C. Zeng, Z. Mu, and X. Gong, "ACPred-BMF: Bidirectional LSTM With Multiple Feature Representations for Explainable Anticancer Peptide Prediction," *Scientific Reports* 12, no. 1 (2022): 21915, <https://doi.org/10.1038/s41598-022-24404-1>.
27. H. Kaleem, S. Rukhsar, and M. N. Khalid, "Anticancer Peptides Prediction: A Deep Learning Approach," *Journal of Computing & Biomedical Informatics* 3, no. 02 (2022): 144–151, <https://doi.org/10.56979/302/2022/81>.
28. L. Zhu, C. Ye, X. Hu, S. Yang, and C. Zhu, "ACP-Check: An Anticancer Peptide Prediction Model Based on Bidirectional Long Short-Term Memory and Multi-Features Fusion Strategy," *Computers in Biology and Medicine* 148 (2022): 105868, <https://doi.org/10.1016/j.combiomed.2022.105868>.
29. Z. Cui, S. G. Wang, Y. He, Z. H. Chen, and Q. H. Zhang, "DeepTPred: A Deep Learning Approach With Matrix Factorization for Predicting Therapeutic Peptides by Integrating Length Information," *IEEE Journal of Biomedical and Health Informatics* 27, no. 9 (2023): 4611–4622, <https://doi.org/10.1109/JBHI.2023.3290014>.

30. V. K. Sangaraju, N. T. Pham, L. Wei, X. Yu, and B. Manavalan, "mACPPred 2.0: Stacked Deep Learning for Anticancer Peptide Prediction With Integrated Spatial and Probabilistic Feature Representations," *Journal of Molecular Biology* 436, no. 17 (2024): 168687, <https://doi.org/10.1016/j.jmb.2024.168687>.
31. N. Schaduagrath, C. Nantasenamat, V. Prachayasittikul, and W. Shoombuatong, "ACPPred: A Computational Tool for the Prediction and Analysis of Anticancer Peptides," *Molecules* 24, no. 10 (2019): 1973, <https://doi.org/10.3390/molecules24101973>.
32. G. Zhong and L. Deng, "Acpsscanner: Prediction of Anticancer Peptides by Integrated Machine Learning Methodologies," *Journal of Chemical Information and Modeling* 64, no. 3 (2024): 1092–1104, <https://doi.org/10.1021/acs.jcim.3c01860>.
33. Q. Yuan, K. Chen, Y. Yu, N. Q. K. Le, and M. C. H. Chua, "Prediction of Anticancer Peptides Based on an Ensemble Model of Deep Learning and Machine Learning Using Ordinal Positional Encoding," *Briefings in Bioinformatics* 24, no. 1 (2023), <https://doi.org/10.1093/bib/bbac630>.
34. P. Agrawal, D. Bhagat, M. Mahalwal, N. Sharma, and G. P. S. Raghava, "AntiCP 2.0: An Updated Model for Predicting Anticancer Peptides," *Briefings in Bioinformatics* 22, no. 3 (2021), <https://doi.org/10.1093/bib/bbaa153>.
35. A. Tyagi, A. Tuknait, P. Anand, et al., "CancerPPD: A Database of Anticancer Peptides and Proteins," *Nucleic Acids Research* 43, no. D1 (2015): D837–D843, <https://doi.org/10.1093/nar/gku892>.
36. Z. Li, J. Fang, S. Wang, L. Zhang, Y. Chen, and C. Pian, "Adapt-Kcr: A Novel Deep Learning Framework for Accurate Prediction of Lysine Crotonylation Sites Based on Learning Embedding Features and Attention Architecture," *Briefings in Bioinformatics* 23, no. 2 (2022), <https://doi.org/10.1093/bib/bbac037>.
37. R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) Neural Networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (IEEE, 2017), 1597–1600, <https://doi.org/10.1109/MWSCAS.2017.8053243>.
38. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation* 9, no. 8 (1997): 1735–1780, <https://doi.org/10.1162/neco.1997.9.8.1735>.
39. Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 1091–1100.
40. R. Xia, Y. Gao, Y. Zhu, D. Gu, and J. Wang, "An Attention-Based Wide and Deep CNN With Dilated Convolutions for Detecting Electricity Theft Considering Imbalanced Data," *Electric Power Systems Research* 214 (2023): 108886, <https://doi.org/10.1016/j.epsr.2022.108886>.
41. P. Zhang, H. Zhang, and H. Wu, "iPro-WAEL: A Comprehensive and Robust Framework for Identifying Promoters in Multiple Species," *Nucleic Acids Research* 50, no. 18 (2022): 10278–10289, <https://doi.org/10.1093/nar/gkac824>.
42. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations (ICLR, 2015)*, <https://doi.org/10.48550/arXiv.1412.6980>.
43. H. Wu, P. Zhang, Z. Ai, et al., "StackTADB: A Stacking-Based Ensemble Learning Model for Predicting the Boundaries of Topologically Associating Domains (TADs) Accurately in Fruit Flies," *Briefings in Bioinformatics* 23, no. 2 (2022), <https://doi.org/10.1093/bib/bbac023>.
44. Z. Yin, J. Lyu, G. Zhang, X. Huang, Q. Ma, and J. Jiang, "SoftVoting6mA: An Improved Ensemble-Based Method for Predicting DNA N6-Methyladenine Sites in Cross-Species Genomes," *Mathematical Biosciences and Engineering* 21, no. 3 (2024): 3798–3815, <https://doi.org/10.3934/mbe.2024169>.
45. Z. Chen, N. He, Y. Huang, W. T. Qin, X. Liu, and L. Li, "Integration of A Deep Learning Classifier With A Random Forest Approach for Predicting Malonylation Sites," *Genomics, Proteomics & Bioinformatics* 16, no. 6 (2018): 451–459, <https://doi.org/10.1016/j.gpb.2018.08.004>.
46. C. W. Tung and S. Y. Ho, "Computational Identification of Ubiquitylation Sites From Protein Sequences," *BMC Bioinformatics* 9, no. 1 (2008): 310, <https://doi.org/10.1186/1471-2105-9-310>.
47. Y. Z. Chen, Z. Chen, Y. A. Gong, and G. Ying, "SUMOhydro: A Novel Method for the Prediction of Sumoylation Sites Based on Hydrophobic Properties," *PLoS One* 7, no. 6 (2012): e39195, <https://doi.org/10.1371/journal.pone.0039195>.
48. B. Rao, C. Zhou, G. Zhang, R. Su, and L. Wei, "ACPred-Fuse: Fusing Multi-View Information Improves the Prediction of Anticancer Peptides," *Briefings in Bioinformatics* 21, no. 5 (2020): 1846–1855, <https://doi.org/10.1093/bib/bbz088>.
49. L. Wei, C. Zhou, R. Su, and Q. Zou, "PEPred-Suite: Improved and Robust Prediction of Therapeutic Peptides Using Adaptive Feature Representation Learning," *Bioinformatics* 35, no. 21 (2019): 4272–4280, <https://doi.org/10.1093/bioinformatics/btz246>.
50. A. Tyagi, P. Kapoor, R. Kumar, K. Chaudhary, A. Gautam, and G. P. Raghava, "In Silico Models for Designing and Discovering Novel Anticancer Peptides," *Scientific Reports* 3, no. 1 (2013): 2984, <https://doi.org/10.1038/srep02984>.
51. L. McInnes, J. Healy, and J. Melville, "Umap: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv preprint arXiv:1802.03426* (2018), <https://doi.org/10.48550/arXiv.1802.03426>.
52. J. B. Ingraham, M. Baranov, Z. Costello, et al., "Illuminating Protein Space With a Programmable Generative Model," *Nature* 623, no. 7989 (2023): 1070–1078, <https://doi.org/10.1038/s41586-023-06728-8>.
53. H. Cui, C. Wang, H. Maan, et al., "scGPT: Toward Building a Foundation Model for Single-Cell Multi-Omics Using Generative AI," *Nature Methods* 21, no. 8 (2024): 1470–1480, <https://doi.org/10.1038/s41592-024-02201-0>.