Supplemental information

TheraPepNet: Enhancing Therapeutic Peptide Recognition via Temporal Sequence Prediction Networks and Preprocessed Features

Jia Tian[1†], Zhenming Wu[2†], Xiaoquan Su[1], Jin Zhao[1*]
[1]School of Computer Science and Technology, Qingdao University, Shandong, China
[2]School of Computer and Information Science, Southwest University, Chongqing, China
[†]These authors contributed equally to this work.
[*]Corresponding author email: zhaojin@qdu.edu.cn

Contents

Supplemental Notes

Supplementary Note 1: Temporal Features

Temporal features refer to the dynamic correlation patterns exhibited by the physicochemical properties of amino acid residues along a peptide sequence. Essentially, they quantify the sequential relationships between adjacent or non-adjacent residues, thereby modeling the dynamic interactions that occur in peptides from the N-terminus to the C-terminus during structural folding, conformational changes, and functional activities.

Supplementary Note 2: The biological significance of CFFT features.

The CFFT features are generated by performing Fourier transform on the hydrophobicity and charge sequences of peptide sequences. At low frequency components, they correspond to the "global distribution trends" of hydrophobicity and charge, and are directly associated with peptide-biological membrane interactions. CPPs achieve penetration through the "insertion of global hydrophobic regions into the cell membrane and binding of local positive charges to negatively charged membrane phospholipids"; the mean value of CFFT hydrophobic low-frequency components of CPPs is significantly higher than that of non-penetrating peptides, and this difference exactly reflects the global structural characteristic of CPPs, i.e., "high hydrophobicity at the N-terminus and positive charge enrichment in the middle region".

At high frequency components, the CFFT features correspond to the "local fluctuation patterns" of hydrophobicity and charge, and are directly related to peptide-receptor binding specificity. AMPs as an example, their bactericidal mechanism relies on the "insertion of local hydrophobic loops into bacterial membrane pores and binding of adjacent positive charges to membrane proteins"; the variance of CFFT charge high-frequency components of AMPs is significantly higher than that of non-antimicrobial peptides, and this fluctuation characteristic exactly matches the local sequence pattern of AMPs, i.e., "alternating hydrophobic loops and positively charged residues".

Supplementary Note 3: Pre-Trained Embedding Vector from ProtBERT

To capture the diverse properties of amino acids, we use pre-trained embedding vectors from ProtBERT, a model designed specifically for protein sequences. ProtBERT is based on the Bidirectional Encoder Representations from Transformers architecture. It learns rich representations through unsupervised learning from large-scale protein sequence data. During pre-training, ProtBERT employs a Masked Language Modeling approach, which helps it capture syntactic and semantic features of sequences.

We obtain the ProtBERT model via the Hugging Face transformers library and use a tokenizer to encode raw peptide sequences into token IDs. To standardize sequence lengths, we apply truncation and padding, while attention masks ensure that the model focuses on real data rather than padding. Finally, we input the processed sequences into the ProtBERT model to extract fixed-dimensional embedding vectors for downstream classification tasks.

Supplementary Note 4: Multi-Head Attention Mechanism

- **Calculation of Queries, Keys, and Values:** Let the input sequence be represented as $X \in \mathbb{R}^{L \times d}$, where L denotes the length of the sequence and d represents the dimensionality of each element's features. For each attention head h, the query, key, and value are obtained by applying linear transformations to the input X using the corresponding weight matrices:

$$Q^{(h)} = XW_Q^{(h)}, \quad K^{(h)} = XW_K^{(h)}, \quad V^{(h)} = XW_V^{(h)} \tag{1}$$

  where $Q^{(h)}$ ($Q^{(h)} \in \mathbb{R}^{L\times d_q}$), $K^{(h)}$ ($K^{(h)} \in \mathbb{R}^{L\times d_k}$), and $V^{(h)}$ ($V^{(h)} \in \mathbb{R}^{L\times d_v}$) represent the query, key, and value vectors, respectively. $W_Q^{(h)}, W_K^{(h)}, and\ W_V^{(h)}$ represent the weight matrices for the queries, keys, and values, respectively.

- **Attention Calculation:** For each attention head h, the attention weights $A^{(h)}$ are first computed by taking the dot product of the query and key, followed by applying the softmax function. The weighted sum of the values is then computed to obtain the output *O*$^{(h)}$:

$$\begin{cases} A^{(h)} = softmax\left(\frac{Q^{(h)}K^{(h)T}}{\sqrt{d_k}}\right) \\ O^{(h)} = A^{(h)}V^{(h)} \end{cases} \tag{2}$$

where $A^{(h)} \in \mathbb{R}^{L\times L}$ and $O^{(h)} \in \mathbb{R}^{L\times d_v}$.

- **Multi-Head Attention Output:** The outputs of all attention heads are concatenated and then passed through a linear transformation to obtain the final multi-head attention output. Specifically, for h attention heads, the multi-head attention output is O $(O \in \mathbb{R}^{L\times hd})$expressed as:

$$O = \left[O^{(1)}, O^{(2)}, \ldots, O^{(h)}\right]W_O \tag{3}$$

where W_O $(W_O \in \mathbb{R}^{hd_v\times d})$ is the linear transformation matrix, and *h* is the number of attention heads.

Supplementary Note 5: Local Feature Computing Network

Given an input peptide sequence *S* of length L , let the convolution kernel $w(w \in \mathbb{R}^K)$ be a one-dimensional kernel of size *K*, and the dilation rate is *d*, such that there are *d*-1 gaps between consecutive elements of the kernel. Therefore, the operation of dilated convolution can be expressed as:

$$Y[t] = \sum_{k=0}^{K-1} w_k \cdot s_{t+d\cdot k}, 0 \leq t \leq L - d \cdot (K-1) - 1 \tag{4}$$

Here, *Y[t]* denotes the value at position *t* in the output sequence, while $s_{t+d\cdot k}$ represents the corresponding value in the input sequence for the *k*-th element of the convolution kernel. The dilation rate *d* controls the spacing between consecutive elements of the kernel.

This contrasts sharply with attention mechanisms, which aggregate global information, allowing each position to depend on data from other positions and thereby capturing global features. Mathematically, causal convolution can be expressed as:

$$Y[t] = \sum_{k=0}^{K-1} w_k \cdot x_{t-k} \cdot \mathbb{I}(t,k), \qquad \mathbb{I}(t,k) = \begin{cases} 1, & t \geq k, \\ 0, & t < k. \end{cases} \tag{5}$$

This mechanism ensures that each output *Y[t]* depends only on the current and past inputs, analogous to how the properties or functions of certain amino acids are determined by the characteristics of their preceding amino acid residues.

Supplementary Note 6: Feed-Forward Network

The Feed-Forward Network (FFN) is typically used to apply non-linear transformations to the representation at each position. It is a fully connected network consisting of two linear transformations and an activation function, with the ReLU function commonly used as the activation. The mathematical expression for the FFN is as follows:

$$Y = W_2 \cdot ReLU(W_1 X + b_1) + b_2 \tag{6}$$

where Y is the output vector, X is the input vector; W_1 and W_2 are the weight matrices for the first and second layers, respectively; b_1 and b_2 are the corresponding bias vectors. These variables work together on the input data to generate the final output Y through linear transformations and the ReLU activation function.

Supplementary Note 7: Gating Mechanism

To dynamically adjust the contribution of each feature output, we introduce a gating mechanism that uses a Sigmoid function to generate gating values between 0 and 1, which are then used to weight the outputs from different modules. The formula for the gating mechanism is presented as follows:

$$\begin{cases} gate = \sigma(w_{gate} \cdot input) \\ output = gate \times input \end{cases} \tag{7}$$

Here, σ denotes the Sigmoid activation function, w_{gate} represents the learnable gating weight matrix, and input refers to the corresponding output of each network module. The gating value for each module controls the extent to which its output is retained. Specifically, when the gating value is close to 1, the output's influence is significant, whereas when the gating value approaches 0, the module's contribution is suppressed.

This gating mechanism effectively modulates the outputs of the temporal network, enhancing the model's expressive power and providing greater control and flexibility for fine-tuning.

Supplementary Note 8: Hadamard product

The selection of the Hadamard product is primarily based on the following three aspects:

1. Dimensional consistency and structural preservation: It fully retains the two-dimensional structure of amino acid physicochemical property indices, avoiding dimensional compression and loss of index-specific information that may occur with other methods.
2. Precise "element-wise weighting" characteristic: The Hadamard product allows all physicochemical indices of each amino acid to be independently "scaled" by their occurrence frequency in the peptide sequence. For instance, when hydrophobic amino acids account for 30% of the sequence in antimicrobial peptides, their 531 physicochemical indices are amplified by 30%, directly enhancing the contribution of "dominant amino acids".
3. Requirement for feature weighting and fusion: TheraPepNet needs to fuse two types of information, namely the AAindex matrix and the AAC vector. As indicated in the reference materials, the Hadamard product is commonly used for feature weighting and fusion in machine learning.

Supplementary Note 9: Assessment Metrics

To assess the performance of the model, we employed commonly used evaluation metrics such as accuracy (ACC), precision (PREC), sensitivity (SEN), specificity (SPE), F1-score, and Matthews correlation coefficient (MCC). These metrics serve as universal indicators for gauging the effectiveness of the model. The detailed description of these six indicators is as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$PREC = \frac{TP}{TP + FP} \tag{9}$$

$$SEN = \frac{TP}{TP + FN} \tag{10}$$

$$SPE = \frac{TN}{TN + FP} \tag{11}$$

$$F1 - score = \frac{2 \cdot PREC \cdot SEN}{PREC + SEN} \tag{12}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{13}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

Supplementary Note 10: Hyperparameter Setting

In this study, the model hyperparameters are set as follows: a batch size of 64, 300 training epochs, a dropout rate of 0.2, and a learning rate of 0.001. For the Global Feature Learning Network, we employed 8 attention heads and a 3-layer network structure to effectively capture global features. The Local Feature Learning Network progressively increases the number of channels ([8, 16, 32, 64, 128]) and uses convolutional kernel with the size of 3 to extract local features. These hyperparameters were carefully tuned to optimize the overall training performance of the model.

Supplementary Note 11: Experimental Setup

TheraPepNet is built on the PyTorch 2.4.1 deep learning framework and developed and tested in a Python 3.8 environment. We utilized four NVIDIA RTX 3090 GPUs for parallel computation to enhance training speed and computational efficiency. Through multi-GPU parallel processing, we were able to significantly accelerate the training process on large-scale datasets. Even without GPU acceleration, the experiments can still run on a CPU.

To reduce the interference of random test set splits on performance metrics, we adopted a scheme of 10 independent repetitions for dataset splitting, validating the model's stability across different data partitions through multiple repetitions. To ensure the precise reproducibility of the experimental results, a fixed random seed of 43 was used for all dataset splitting processes.

Supplementary Note 12: Datasets

For detailed information on the six therapeutic peptides, please refer to Table I. The characteristics of the aforementioned datasets, including subtype classification and quantitative annotation, align closely with the core tasks of this study—therapeutic peptide multi-class fine classification and the association of sequence, function, and mechanism. These characteristics are superior to alternative datasets with similar small sample sizes and low annotation quality.

Table S1. Characteristics of benchmarking datasets.

dataset	type	positive	negative	total
ABP	antibacterial peptide	800	800	1600
AIP	anti-inflammatory peptide	1887	1258	3145
AVP	antiviral peptide	407	544	951
CPP	cell penetrating peptide	370	370	740
PBP	polystyrene surface binding peptide	80	80	160
QSP	quorun sensing peptide	200	200	400

Supplementary Note 13: Reasons for TheraPepNet's advantages

The fundamental reason for the improved accuracy of TheraPepNet lies in its architectural design, where CFFT+GFN precisely matches the biological essence of the peptide structure-function relationship. Additionally, through a dynamic weighted integration strategy, it flexibly balances the feature requirements across different categories, specifically addressing the core limitations of traditional methods, such as the lack of structural features and rigid integration. This ultimately leads to a significant enhancement in classification performance.

Supplemental Figures

1. Multi Classification

In this study, we combined functional peptide datasets CPP, PBP, and QSP to create a sample set containing peptide sequences from multiple categories. The dataset was then split into training, validation, and test sets to assess the performance of TheraPepNet in functional peptide recognition. Figure S1. illustrates the performance of TheraPepNet on the independent test set. The clusters before model training are highly overlapping; the clusters after TheraPepNet training have clear boundaries.

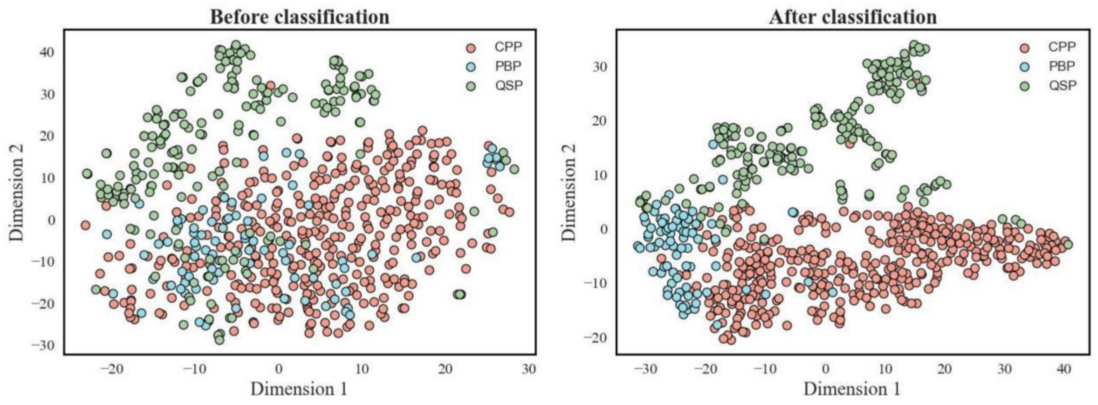

Figure S1. t-SNE dimensionality reduction was applied to visualize the distribution of samples in the two-dimensional space before and after classification. Three types of functional therapeutic peptides, CPP, PBP, and QSP, were employed in the multi-class classification task.

2. Feature Contribution

By adjusting the hyperparameters, we set the output vectors of each feature combination to 128 dimensions, resulting in six combinations: CFFT+GFN, CFFT+LFN, ProtBERT+GFN, ProtBERT+LFN, AAindex+AAC+GFN, and AAindex+AAC+LFN, yielding feature vectors of length 768. After forward and backward propagation, we extracted the weights from the final fully connected layer to evaluate the impact of each input feature on the output. Contribution scores were calculated by averaging the weights across output neurons and normalized to form a probability distribution, as shown in Figure S2.

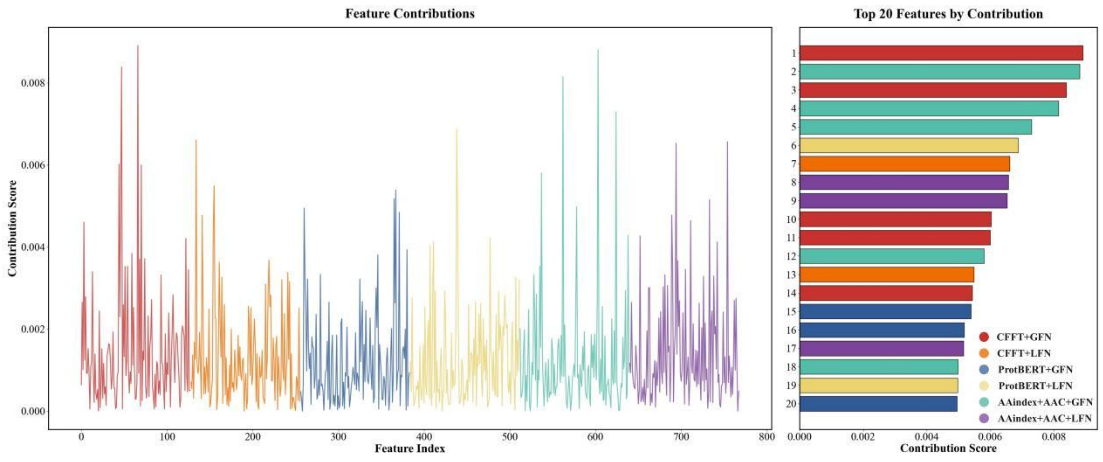

Figure S2. Analysis of feature contributions in the TheraPepNet model. The left panel shows the contribution scores of six features across the integrated CPP, PBP, and QSP datasets. The right panel displays the top 20 ranked features.

CFFT+GFN and AAindex+AAC+GFN emerged as pivotal feature combinations. The top 20 features, along with their index sources, revealed that 5 features came from CFFT+GFN, 5 from AAindex+AAC+GFN, 3 from AAindex+AAC+LFN, and 3 from ProtBERT+GFN. Only 2 features were derived from CFFT+LFN and ProtBERT+LFN. Notably, features from CFFT+GFN and AAindex+AAC+GFN each accounted for a quarter of the total contributions, underscoring their significant impact on classification performance.

The feature importance of CFFT+GFN is significantly higher than that of other combinations, primarily because this combination precisely matches the biological patterns of peptide function dependent on secondary structural characteristics. For α-helix-dominated CPPs/AMPs, the model primarily relies on the low-frequency periodic components of CFFT and the hydrogen bonding information from GFN to identify helical morphology and assess helix stability. For β-sheet-dominated AIPs/defensins, the model prioritizes the high-frequency fluctuation components of CFFT and the receptor binding weights from GFN to identify residue pairings and screen for functionally relevant folding regions. This precise match between "structural features and functional requirements" makes CFFT+GFN the core feature for the model in discriminating peptide function, also explaining its dominant biological essence in the importance scores.

3. Lengths Distribution

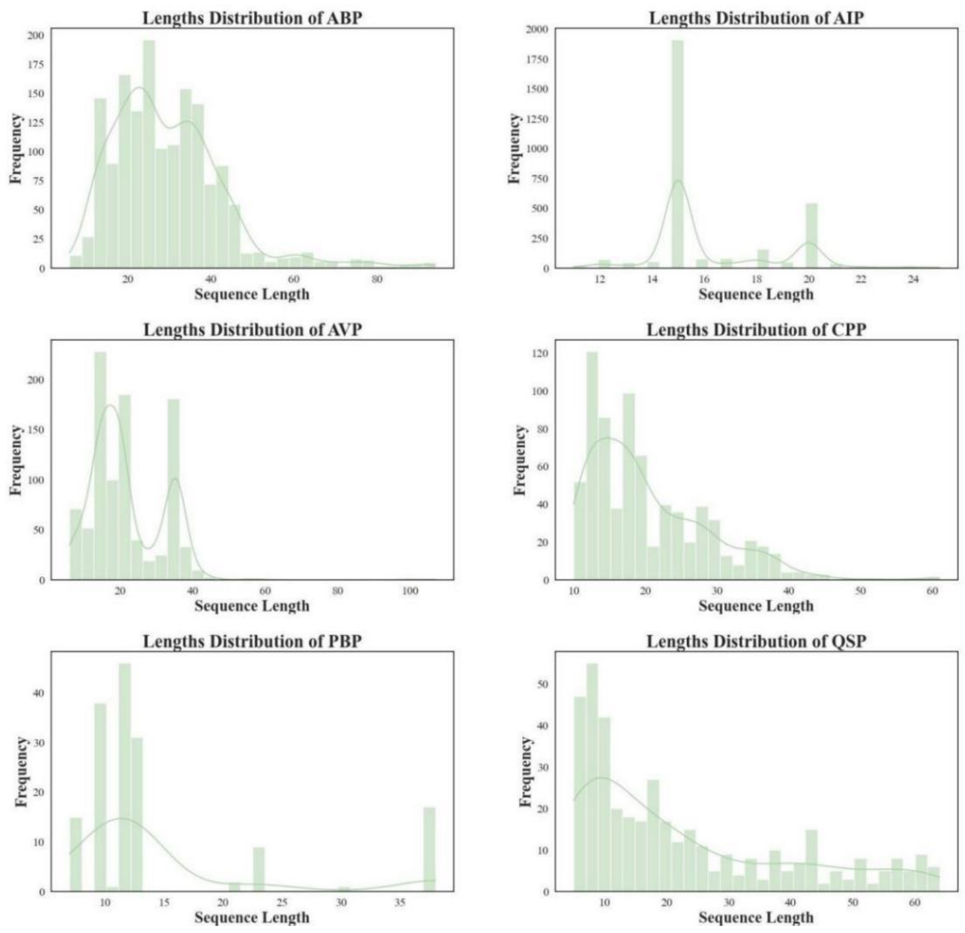

Figure S3. Distribution of peptide sequence lengths across six datasets: ABP, AIP, AVP, CPP, PBP, and QSP.

4. Hydrophobicity and Charge Distribution

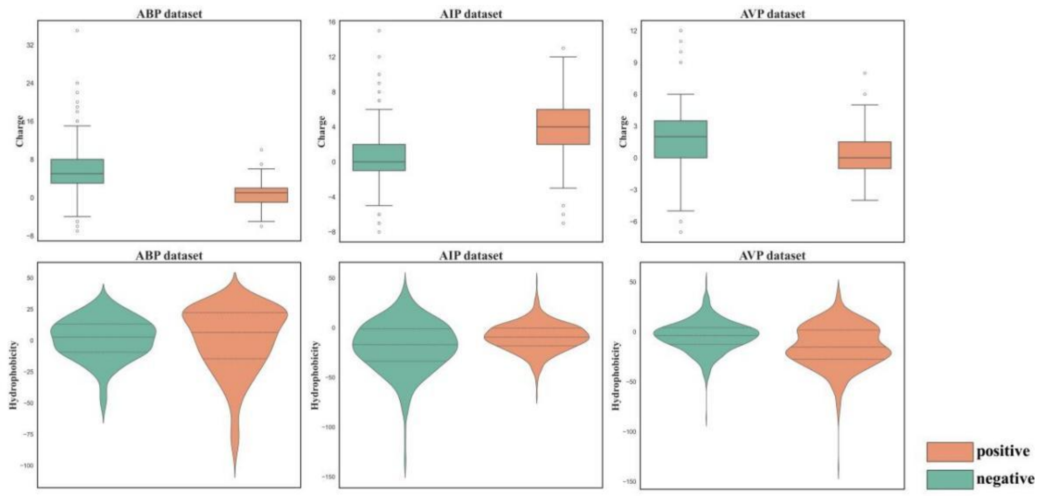

Figure S4. The hydrophobicity and charge distribution characteristics of positive and negative samples in the ABP, AIP, and AVP datasets.

5. Learning Curve

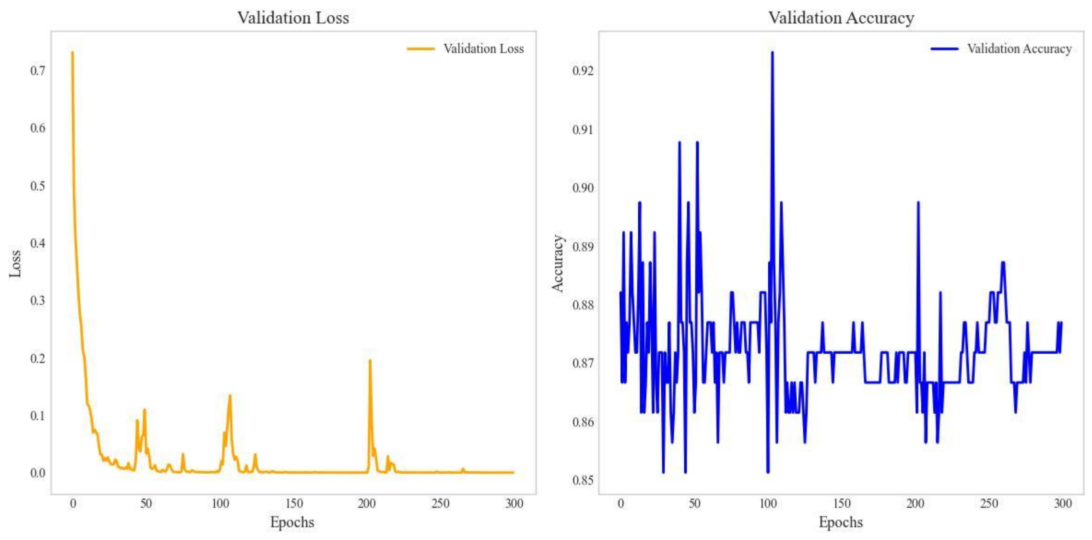

Figure S5. Learning curve of TheraPepNet on the mixed dataset composed of CPP, PBP, and QSP. The changes in the validation loss and accuracy are shown as the number of epochs increases.

Supplemental Tables

1. Experimental Results

Table S2. The Performance of TheraPepNet and Four Other Methods on the ABP.

Method	ACC	SE	SP	F1-score	MCC
ETFC	0.894	0.929	0.873	0.879	0.803
PrMFTP	0.903	0.910	0.895	0.899	0.806
TPpred-ATMV	0.911	0.852	0.873	0.866	0.826
DeepTPpred	0.915	0.904	0.876	0.886	0.822
TheraPepNet	0.929	0.932	0.927	0.930	0.860

Table S3. The Performance of TheraPepNet and Four Other Methods on the AIP.

Method	ACC	SE	SP	F1-score	MCC
ETFC	0.613	0.631	0.592	0.687	0.220
PrMFTP	0.675	0.610	0.661	0.705	0.290
TPpred-ATMV	0.639	0.690	0.662	0.703	0.251
DeepTPpred	0.648	0.682	0.586	0.710	0.260
TheraPepNet	0.707	0.681	0.716	0.698	0.313

Table S4. The Performance of TheraPepNet and Four Other Methods on the AVP.

Method	ACC	SE	SP	F1-score	MCC
ETFC	0.800	0.790	0.808	0.768	0.595
PrMFTP	0.837	0.763	0.800	0.766	0.675
TPpred-ATMV	0.767	0.719	0.762	0.725	0.536
DeepTPpred	0.821	0.775	0.857	0.805	0.635
TheraPepNet	0.854	0.809	0.891	0.861	0.705

2. Consuming Time and Memory

Table S5. GPU Time for Running 300 Epochs Across Different Datasets for Various Methods (Unit of Measurement: Seconds).

method dataset	ABP	AIP	AVP
ETFC	516.0s	879.2s	203.4s
PrMFTP	447.0s	719.2s	274.6s
TPpred-ATMV	106.5s	256.9s	65.2s
DeepTPpred	75.3s	97.8s	76.4s
TheraPepNet	281.1s	428.3s	183.4s

Table S6. Computational Memory Requirements of Different Methods on Various Datasets (Unit of Measurement: MegaByte).

method dataset	ABP	AIP	AVP
ETFC	553.37 MB	551.55 MB	552.52 MB
PrMFTP	359.88 MB	422.52 MB	392.22 MB
TPpred-ATMV	349.50 MB	353.44 MB	349.27 MB
DeepTPpred	356.67 MB	357.07 MB	357.67 MB
TheraPepNet	334.00 MB	320.78 MB	339.60 MB