Analytical Biochemistry 705 (2025) 115920

Contents lists available at ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio

NeuroCL: A deep learning approach for identifying neuropeptides based on contrastive learning

Jian Liu [a], Aoyun Geng [a], Feifei Cui [a], Junlin Xu [b], Yajie Meng [c], Leyi Wei [d,e], Qingchen Zhang [a], Quan Zou [f], Zilong Zhang [a,*]

[a] School of Computer Science and Technology, Hainan University, Haikou, 570228, China
[b] School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, 430081, Hubei, China
[c] School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, 430200, Hubei, China
[d] Centre for Artificial Intelligence Driven Drug Discovery, Faculty of Applied Science, Macao Polytechnic University, Macao Special Administrative Region of China
[e] School of Informatics, Xiamen University, Xiamen, China
[f] Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, 610054, China

ARTICLE INFO

Keywords:
Deep learning
Contrastive learning
Neuropeptide prediction
Cross-attention mechanism

ABSTRACT

Neuropeptides (NPs), a unique class of neuronal signaling molecules, involved in neurotransmission, endocrine regulation, immune response, mood, and appetite control. The identification of neuropeptides provides critical scientific insights for early diagnosis, targeted therapy, and personalized medicine of related diseases. Previous models struggle to capture complex relationships among features and inter-sample connections. In this study, we introduce NeuroCL, a deep learning model harnessing contrastive learning and a cross-attention mechanism to efficiently identify NPs through multifaceted attribute representation. Experimental outcomes demonstrate that NeuroCL effectively captures data nuances, achieving an impressive accuracy of 93.8 % and a Matthews correlation coefficient (MCC) of 87.8 % on an independent test set. Contrastive learning enhances class distinction and coherence, while cross-attention mechanisms integrate pre-trained large models with manually encoded features, synergistically boosting their capabilities and strengthening feature connections. Our model surpasses current state-of-the-art predictors in NPs identification. Visualization via uniform manifold approximation and projection (UMAP) reveals that NeuroCL distinctly segregates positive NPs from negative ones. To facilitate the accessibility and application of our model, we have established a web-based platform available at http://www.bioai-lab.com/NeuroCL.

1. Introduction

Neuropeptides (NPs), synthesized by neurons, function as signaling molecules within neurotransmission via interaction with G-protein-coupled receptors (GPCRs) to exert their effects. These molecules range from 3 to 100 amino acids in length and represent the most versatile neurotransmitters in the neuroimmune system. Derived from larger precursor proteins containing distinctive C-terminal or N-terminal biochemical signatures [1–3], NPs crucially regulate pathological states, including substance addiction, neural injury, stress responses, and nociception [4]. As molecular mediators, they orchestrate neurotransmission and physiological homeostasis across phylogenetically diverse species [5,6], with essential roles in growth regulation, neuromuscular function, circadian rhythms, metabolic processes, cognitive functions, and affective states [7]. Particularly in neurodegenerative contexts, NPs serve as key neuromodulators of synaptic plasticity, with dysregulation pathologically implicated in Parkinson's disease progression [8,9]. Within the neural system, they function as neurotransmitters, neuromodulators, or neurohormones [10–13], and are acknowledged as crucial therapeutic tools for neurological disorders and lead compounds for novel drug development.

Discovering neuropeptides is crucial for deciphering neurophysiological mechanisms, as well as for rational drug discovery. While traditional experimental methods enable accurate identification of NPs, their practical utility is limited by time-intensive workflows, high operational costs, and stringent demands for high-quality datasets [14,15]; thus, computational predictive models offer a more feasible approach for identifying neuropeptides. Bioinformatics has seen

[*] Corresponding author.
E-mail address: zhangzilong@hainanu.edu.cn (Z. Zhang).

https://doi.org/10.1016/j.ab.2025.115920
Received 16 March 2025; Received in revised form 13 May 2025; Accepted 1 June 2025
Available online 2 June 2025

revolutionary integration of machine learning algorithms [16], leading to the development of numerous computational approaches utilizing supervised machine learning frameworks for identifying neuropeptides or their precursor molecules [17–19]. Agrawal et al. introduced NeuroPIpred, a computational technique driven by machine learning, to establish a support vector machine (SVM) [20] model optimized for predicting neuropeptides in insects [21]. Advances in machine learning have enabled the emergence of a multitude of methodologies within this field, necessitating integrative strategies rather than relying solely on a single method. Ensemble learning-based approaches have demonstrated superior predictive performance. In 2020, Bin et al. pioneered PredNeuroP, a two-layer stacked ensemble model for neuropeptide prediction [17], leveraging the extensive NeuroPep database [22]. This model implemented consensus diverse feature selection algorithms to curate distinct feature sets for training its underlying learners. During 2021, methodological breakthroughs occurred in ensemble learning techniques for neuropeptide prediction. Jiang et al. introduced an interpretable stacked model that integrates diverse sequence features and feature selection approaches for neuropeptide prediction [18]. Additionally, Hasan et al. presented NeuroPredFRL, an efficient predictor that utilizes feature representation learning methods [23].

The progression of deep learning in recent years has revolutionized computational methodologies for enhancing predictive model capabilities [24–26]. The emergence of large language models tailored for proteins, such as Evolutionary Scale Modeling (ESM-2), has catalyzed transformative paradigm shifts in bioinformatics, enabling automated feature extraction from raw amino acid sequences without laborious manual feature engineering [27,28]. This contrasts sharply with traditional machine learning approaches that depend on handcrafted descriptors (e.g., amino acid composition, physicochemical properties), which often inadequately capture sequence complexity. Numerous models leveraging bidirectional gated recurrent units (BiGRUs) and recurrent neural networks (RNNs) have demonstrated accurate predictions for TAP-binding peptides [29]. Ahmed et al. proposed a combined CNN-BiLSTM-Attention model to predict antimicrobial peptides [30,31].

Experimental validation through techniques like mass spectrometry remains constrained by prohibitively expensive equipment costs and complex sample preprocessing workflows. While conventional methods persist in neuropeptide research through naive feature concatenation approaches, they fundamentally neglect cross-sample semantic correlations critical for biological pattern discovery. Conventional machine learning approaches depend on handcrafted features [32,33], including amino acid composition and physicochemical attributes, which might not adequately grasp the intricate sequence relationships and require substantial expert knowledge and time investment. Notably, advanced deep learning approaches including self-attention mechanisms and contrastive learning frameworks remain underexplored in this domain, despite their proven success in related fields like medical image analysis and few-shot learning scenarios.

Based on the above aspects, we developed an innovative computational framework combining transformer architectures with advanced learning paradigms. We leveraged the ESM-2 transformer model, which demonstrates state-of-the-art performance in biological sequence processing through its 650M-parameter design. By integrating contrastive learning and cross-attention mechanisms, our model is capable of more effectively capturing the complex relationships between sequences and focusing on key biological features. Compared to existing methods such as NeuroPpred-SVM and NeuroPpred-Fuse, our model demonstrated superior accuracy and AUROC on an independent test set. Moreover, the ESM-2 model exhibits improved biological interpretability, offering profound insights into neuropeptide identification. This study introduces a novel approach to computational neuropeptidomics, presenting a scalable prediction framework along with mechanistic understandings that further our knowledge of peptide signaling networks.

2. Materials and methods

The model's workflow is illustrated and comprises four primary stages: data acquisition (Fig. 1A), amino acid embedding encoder and decoder (Fig. 1B), contrastive learning and classifier (Fig. 1C), and cloud-based web application interface (Fig. 1D).

2.1. Datasets

Model performance fundamentally depends on dataset quality and diversity [34]. To conduct a fair comparison of our model with existing ones, we employed the dataset that was sourced from the SwissProt and NeuroPep databases, which comprises 2425 experimentally confirmed neuropeptides (NPs) and an equal number of non-neuropeptides (non-NPs). NPs were obtained from NeuroPep database, a dedicated resource for NPs, while the non-neuropeptides (non-NPs) sequences were sourced from the SwissProt database. To prevent inaccurate experimental outcomes resulting from excessive similarity among samples, we employed CD-HIT to remove sequences with over 90 % similarity. The dataset was subsequently divided into a training set, which comprised 80 % of the total data, and a test set that included the remaining 20 %. This test set was used to evaluate and contrast the performance of different models [35] (Table 1). We utilized a 10-fold cross-validation approach for hyperparameter tuning and model selection, which not only helps prevent overfitting but also enhances the stability and reliability of the model performance. Sequence length distributions exhibited highly similar profiles between cohorts, suggesting consistent data distributions that may support model applicability to sequences spanning 0–100 residues (Fig. 2).

2.2. Feature representation of peptides

Model performance critically depends on robust sequence representations derived from amino acid residues [36]. To extract key information, we have applied eight peptide sequence encoding strategies: ordinal encoding, position-specific amino acid composition, dipeptide composition, amino acid composition, adaptive skip dipeptide composition [37], amino acid entropy, BLOSUM62 and features extracted by the pre-trained large model ESM-2.

2.2.1. Ordinal encoding

Firstly, we categorize amino acids into 21 distinct groups, comprising the 20 canonical amino acids along with other amino acids, and assign a specific code to each category. To ensure data consistency, we pad the sequences with zeros at the end for sequences with a length less than 100.

2.2.2. Amino acid composition

Amino acid composition (AAC) provides statistical characterization of protein sequences by quantifying the normalized frequency of occurrence of each amino acid type in a given peptide sequence [38,39]. The logic for calculating this 21-dimensional descriptor is as follows:

$$f(i) = \frac{N(i)}{L}, i \in \{A, C, D, \ldots, Y\} \tag{1}$$

where $f(i)$ represents the normalized frequency of amino acid i [40], $N(i)$ is the sum of how often each natural amino acid i occurs in the peptide sequences, and the total length of the peptide chains is denoted by L.

2.2.3. Dipeptide composition

Dipeptide composition (DPC) extends the amino acid composition (AAC) analysis to include dipeptide frequencies. This provides deeper insight into the local interactions that are crucial for peptide structure and function. The calculation of DPC is outlined as follows:

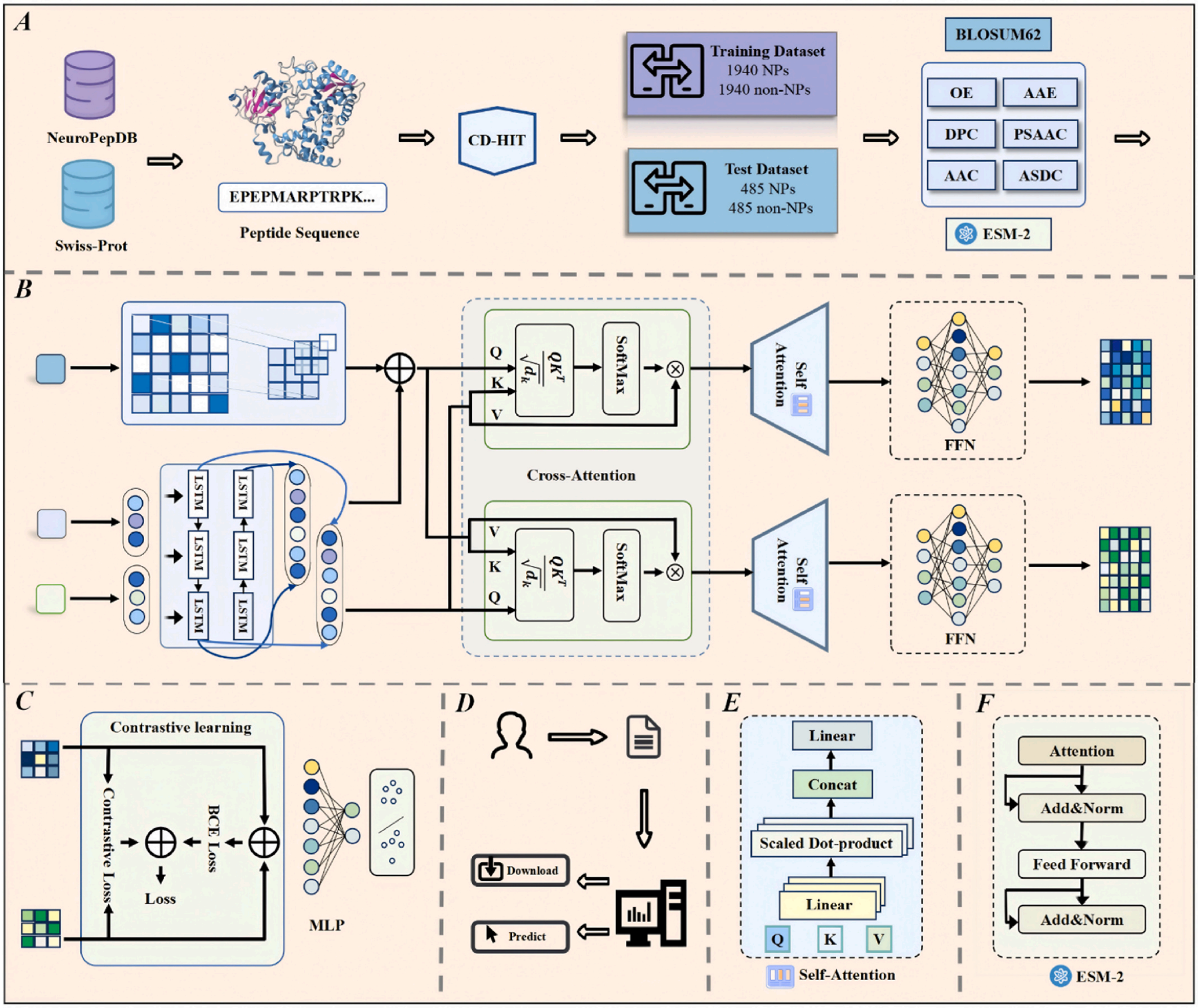

Fig. 1. The overall architecture of NeuroCL. (A) Dataset sources: The dataset is sourced from two databases, and CD-HIT is applied to eliminate redundant sequences, resulting in a refined dataset for training and testing. (B) Encoder and decoder architecture: Input sequences are processed through various encoding techniques, with features being extracted through cross-attention and self-attention mechanisms. Subsequently, a feed-forward neural network decodes these features into the final representations. (C) Contrastive learning and classification architecture: Features from two distinct representations are contrasted. The refined features are then subjected to classification by an MLP. (D) Web server interface: Users can upload their sequence files to obtain probabilistic prediction results. (E) The architecture of self-attention: This mechanism captures long-range dependencies by computing correlations using Q, K and V. (F) The architecture of ESM-2: Built on the Transformer architecture, this module employs an efficient feature extraction method utilizing a multi-head attention mechanism.

Table 1
Brief overview of the applied datasets.

Datasets	Positive	Negative	Total
Training Set	1940	1940	3880
Test Set	485	485	970

$$f(i,j) = \frac{N_{ij}}{L-1}, i,j \in \{A, C, D, \ldots, Y\} \tag{2}$$

where $\mathrm{f}(\mathrm{i},\mathrm{j})$ is the frequencies of dipeptides composed of amino acids; The count of each dipeptide composed of amino acid types i and j is noted by N_{ij}, with L representing the overall sequence length of amino acids.

2.2.4. Position-specific amino acid composition

Given the crucial role of terminal residues in determining the structure and functionality of bioactive peptides, the PSAAC was calculated exclusively for the initial five N-terminal amino acids (NT5) and the final five C-terminal amino acids (CT5) of each peptide segment in the training dataset [21], as outlined subsequently:

$$f(r,i) = \frac{N(r,i)}{M} \tag{3}$$

where $N(r,i)$ signifies the frequency of occurrence of residue r at position i, while M represents the count of peptides within the training dataset. Consequently, we derived two 20 × 5 matrices, each capturing position-specific residue propensities. For any given peptide, its PSAAC was derived by populating these matrices based on the residue type and position i for both the NT5 and CT5 residues.

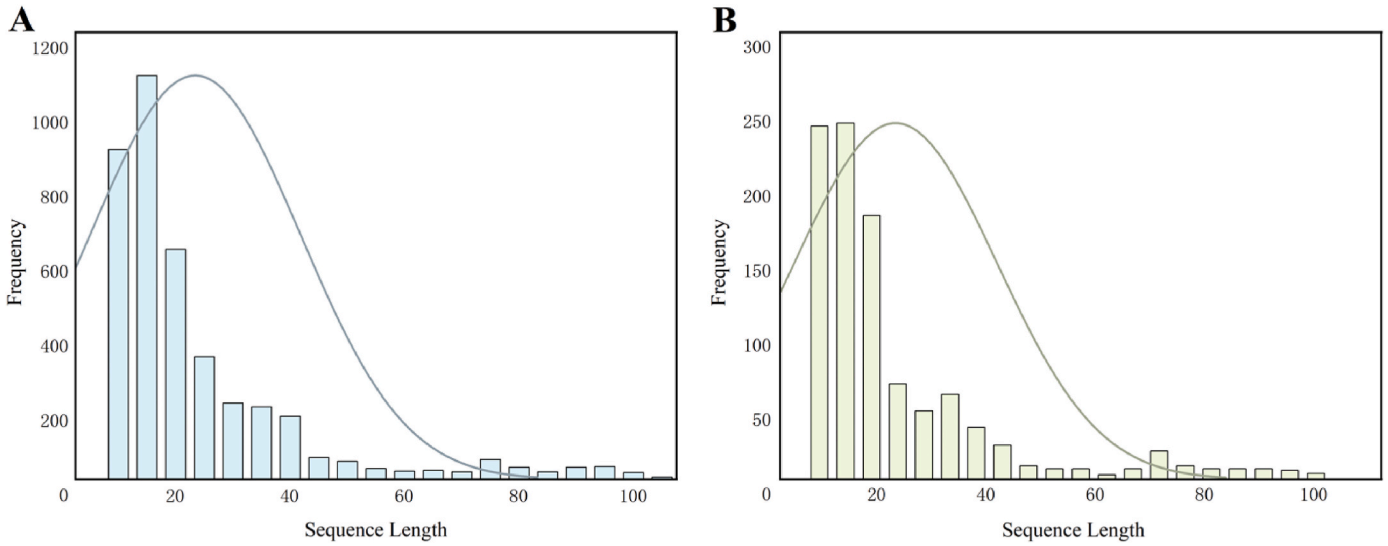

Fig. 2. Peptide Length Distribution of NPs. (A) Length distribution of NPs in the training set. (B) Length distribution of NPs in the test set.

2.2.5. *Adaptive skip dipeptide composition*

The adaptive skip dipeptide composition (ASDC) represents an enhanced variant of the skip-*n*-gram model [17]. This model accommodates the diverse skip distance within the dataset and integrates more comprehensive distance data into its feature set. The ASDC can be derived as outlined below.

$$FV = \left\{ \frac{N(a_{m_1} a_{m_2} \dots a_{m_n})}{N(T_{skipgram})} \right\} \tag{4}$$

$$T_{skipgram} = U_{a-1}^{k} Skip(DT = a) \tag{5}$$

where $Skip(DT = a) = A_i A_{i+a+1} \dots A_{i+a+n-1} \mid 1 \leq i \leq L - a, 1 \leq a \leq k\}$. The approach considers the n residues situated within distances ranging from 1 to k, where k signifies the length of each individual sequence.

2.2.6. *Amino acid entropy*

The amino acid entropy (AAE) embodies the principle of molecular information entropy, reflecting the uneven distribution of each amino acid throughout entire sequences, as well as within the N-terminal five segments (NT5) and C-terminal five segments (CT5) [17]. For a specified peptide sequence P, the entropy rating for amino acid A is encoded as outlined below.

$$AAE = \sum_{i=1}^{p+1} \left(\frac{s_i - s_{i-1}}{n} \right) \log_2 \left(\frac{s_i - s_{i-1}}{n} \right) \tag{6}$$

where *n* signifies the count of amino acid A within peptide P while *p* denotes the length of the peptide P [41]; The positions of A within the sequence are indicated by indices such as $s_1, \dots,$ and s_p, with the initial position defined as $s_0 = 0$ and the final position as $s_{p+1} = n + 1$.

2.2.7. *BLOSUM62*

The BLOSUM62 matrix plays a vital role in bioinformatics, serving as an essential instrument for aligning peptide sequences, and it is constructed based on observed amino acid substitution frequencies. In the BLOSUM62 matrix, the score $S(a,b)$ between any two amino acid residues *a* and *b* is precisely calculated using a log-odds ratio equation.

$$S(a,b) = \frac{1}{\delta} \log \frac{p_{ab}}{f_a f_b} \tag{7}$$

where p_{ab} denotes the target probability, which signifies the occurrence of residue pair *a* and *b* in an alignment with homologous sequences when homology is present. In contrast, $f_a f_b$ represents the mean probability of residue pairs *a* and *b* appearing in any peptide amino acid sequence, assuming they are not homologous. The parameter δ serves as a scaling factor in this context.

2.2.8. *Feature extraction based on the protein language model*

The advent of pre-trained protein language models has catalyzed a paradigm shift in computational biology, enabling hierarchical feature extraction through self-supervised learning on terascale sequence datasets [42]. By analyzing vast amounts of unlabeled protein sequences, these models can acquire representations of protein sequences, thereby significantly enhancing the performance of downstream tasks. Incorporating pre-trained language models can diminish the likelihood of overfitting when confronted with limited training datasets, serving as an effective regularization strategy. In this study, we utilize a self-supervised language model named ESM-2, utilizing the transformer framework and featuring more than 650 million parameters, comprising 33 transformer layers. The ESM-2 model can take a protein sequence as input and output a dynamic embedding vector with a dimension of L × 1280, where L represents the length of the protein chain [43]. It is important to note that we employed the pre-trained ESM-2 model directly for feature extraction without any fine-tuning. This approach not only helps us avoid substantial computational costs but also capitalizes on the robust generalization capability of the pre-trained ESM-2 model. By directly employing it, we can effectively mitigate the risk of overfitting that could otherwise arise from the relatively small size of our specific dataset.

2.3. *Model architecture*

2.3.1. *Multi-scale convolutional neural networks*

In our research, we employed the architecture of a multi-scale convolutional neural network (multi-scale CNN) due to its exceptional performance in processing sequence data [44,45]. This architecture is capable of capturing multi-level features by utilizing convolutional kernels of different sizes. Specifically, we introduced five different sizes of convolutional kernels to process the encoded amino acid sequences, allowing for the consideration of both local details and global contexts, which is crucial for understanding the complex interactions of amino acids within the sequence [12]. For a given input sample X, the mathematical expression for the convolutional layer can be represented as:

$$y_i^k = Relu\left(\sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} w_{pq}^k \cdot x_{i+p,q} + b^k \right) \tag{8}$$

where $Relu(x) = max(0, x)$. w^k is the weight matrix of the k-th convolutional kernel, with dimensions $P \times Q$.

In our model, given the substantial rise in feature map dimensions resulting from the convolution operation, we adopt the max pooling strategy to derive a more concise and simplified feature representation. Max pooling is an effective downsampling technique that diminishes the spatial extent of feature maps by identifying and retaining the maximum value within specified local regions, or receptive fields, while preserving the most significant feature information. This method leads to a reduction in model parameters and computational cost, while simultaneously improving the model's robustness to input variations. For a given input sample X, the convolutional layer can be mathematically represented as follows:

$$pooling(X)_{i,k} = \max\left(X_{iM,k}, X_{iM+1,k}, \ldots, X_{iM+M-1,k}\right) \tag{9}$$

where X denotes the input feature matrix, i signifies the row position within the matrix, k corresponds to the column index of distinct feature channels, and M represents the size of the pooling window used in the operation.

2.3.2. BiLSTM module

The widespread application of recurrent neural networks (RNNs) across numerous domains stems from their remarkable capability to capture temporal information in sequential data. Nevertheless, when dealing with extended sequences, traditional RNNs frequently confront the issue known as the "vanishing gradient" problem, which limits their effectiveness in capturing long-term dependencies. To tackle this issue, long short-term memory networks (LSTM) were proposed and widely used in practice [46]. By applying LSTM both forward and backward on the input sequence [47], bidirectional LSTM (BiLSTM) can capture more comprehensive contextual information and effectively handle long sequences while maintaining long-term information propagation within the sequence through the introduction of unique gating mechanisms [48]. The core formulas of LSTM are:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{10}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{11}$$

$$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{12}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{13}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{14}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{15}$$

At time t, the activation vectors for the forget gate, input gate, and output gate are denoted as f_t, i_t and o_t respectively, and they function to control the transmission of sequential information. The candidate cell state ($\widetilde{C}_t$) and the actual cell state (C_t) serve as indicators of the network's memory. The hidden state vector (h_t) which encapsulates the information processed up to time t, provides a summary of the data.

The fundamental components of the LSTM, including weight matrices (W_f, W_i, W_C, W_o) and bias vectors (b_f, b_i, b_C, b_o), facilitate gate operations and memory updates through the utilization of the sigmoid function σ and the hyperbolic tangent function tanh, which serves to regulate the activations and updates pertaining to the cell states. Within the BiLSTM framework, results are generated from both the forward and backward passes, represented by $h_t^{forward}$ and $h_t^{backward}$ respectively. Ultimately, the concatenation process is employed to merge the forward and backward hidden states together.

2.3.3. Cross-attention mechanism

The cross-attention mechanism has gained widespread adoption in deep learning, especially within sequence-to-sequence architectures and transformer-based models [49]. It primarily serves for information interaction between different inputs, enabling the model to effectively align and focus on contexts from different sources, thereby helping the model better capture the correlations between two inputs.

Given two input sequences $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$, the cross-attention mechanism calculates a set of attention weights A, where each element a_{ij} represents the importance of the i-th element x_i to the j-th element y_j. The attention weights are calculated using the following formula:

$$A = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{16}$$

where Q is the query vector derived from the sequence Y, K is the key vector derived from the input sequence X, and d_k is the dimensionality of the key vector, which is used to scale the dot product to prevent gradients. The attention weights A are then used to create a weighted sum of the sequence X, which is used to compute the output sequence:

$$V = A \cdot X \tag{17}$$

where V is the weighted sum of the sequence X, and each element v_j is computed as:

$$v_j = \sum_{i-1}^{m} a_{ij} x_i \tag{18}$$

The weighted sum V serves as the input to a feed-forward neural network that produces the output sequence. By adopting this approach, two new representations of sequences are generated. During the production of each part of the output sequence, the cross-attention mechanism grants the model the ability to shift its focus seamlessly across diverse segments of the input sequence, which is particularly useful for handling the variable-length and complex relationships between input and output in tasks like machine translation. By incorporating cross-attention, models can better capture the dependencies between input and output sequences, leading to improved performance on a variety of sequence-to-sequence tasks.

2.3.4. Multi-head attention mechanism

The transformer architecture features multi-head attention as a pivotal element. The input information is split into several subspaces by the multi-head attention mechanism. It calculates attention weights independently within each subspace and concatenates the outputs from each head to form the final output. This mechanism not only increases the capacity of the model but also enables it to capture feature information of different dimensions in the input data [50]. The computation of multi-head attention is performed according to the following formula:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{19}$$

where Q denotes the matrices of queries, K signifies the matrices of keys, V represents the matrices of values.

2.3.5. Binary cross-entropy loss function

We employed binary cross-entropy (BCELoss) as the loss function, which is defined as follows:

$$BCELoss = \frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \tag{20}$$

where N is the number of training samples; y_i is the actual label of the samples; p_i is the predicted output of the model for the sample.

2.3.6. Contrastive loss function

The creation of a feature space, where similar samples are brought closer together and dissimilar samples are pushed farther apart, is at the core of contrastive learning. During training, we employed a contrastive loss function to achieve this objective [51]. For any two samples, x_1 and x_2 in the dataset, their similarity is indicated by a binary label y: $y = 0$ signifies similarity, while $y = 1$ signifies dissimilarity. The distance between these two samples in the feature space is quantified using the Euclidean distance. In our study, x_1 and x_2 specifically refer to the embeddings of pre-trained large models and handcrafted features after they have been processed by the encoder and decoder, respectively.

$$D_w(x_1, x_2) = ||x_1 - x_2||_2 \tag{21}$$

The contrastive loss function consists of two key parts: When samples are similar (when $y = 0$), the component known as the loss for positive pairs aims to reduce the distance between them in the feature space. In mathematical terms, this is expressed as:

$$L_{positive} = (1 - y) \times D_w(x_1 - x_2)^2 \tag{22}$$

Conversely, for dissimilar samples (when $y = 1$), the loss, referred to as the loss for negative pairs, strives to maintain a minimum predefined distance, known as a ‘margin’ between them. Mathematically, this is expressed as:

$$L_{negative} = y \times \max\left(0, margin - D_w(x_1, x_2)\right)^2 \tag{23}$$

Combining these components results in the total contrastive loss:

$$L_{contrastive} = L_{positive} + L_{negative} \tag{24}$$

Additionally, the loss is determined by averaging it across all sample pairs contained within a batch.

$$L = \frac{1}{N}\sum_{i=1}^{N} L_{contrastive} \tag{25}$$

2.4. Hyperparameter optimization

The learning rate was systematically evaluated through grid search across a range of values ({0.0001, 0.0002, 0.0004, 0.0006, 0.0008, 0.001}), with the optimal value of 0.0008 selected based on the stability of validation loss convergence. Similarly, the batch size was optimized through grid search across values of {16, 32, 64, 128}, with 32 selected as the optimal value. The Adam optimizer was employed as the primary optimization algorithm due to its efficiency and adaptability. These hyperparameters were chosen to maximize model performance and stability. We used grid search to systematically evaluate different combinations of hyperparameters. For each combination, we trained the model and evaluated its performance on the validation set. The hyperparameters that yielded the best validation performance were selected for the final model.

2.5. Evaluation metrics

The model's efficacy was thoroughly assessed utilizing multiple well-established evaluation metrics, critical for evaluating classification models in bioinformatics [52], including Matthews correlation coefficient (MCC), sensitivity (Sn), specificity (Sp), the area under the curve (AUC), and accuracy (ACC) [53,54]. The calculation of these metrics is performed:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

$$Sn = \frac{TP}{TP + FN} \tag{27}$$

$$Sp = \frac{TN}{TN + FP} \tag{28}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{29}$$

where, TP, FP, TN and FN denote the counts of true positives, false positives, true negatives, and false negatives, respectively, pertaining to the sample classification. MCC was prioritized as the primary metric due to its robustness to class imbalance, particularly crucial for biological datasets with skewed sample distributions. In addition, we employed the area under the receiver operating characteristic curve to quantify model discriminative capacity. This metric integrates the relationship between true positive rates and false positive rates across all classification thresholds, where a higher AUROC value indicates superior ability to distinguish positive NPs from negative ones.

3. Results and discussion

3.1. Systematic performance benchmarking of diverse deep learning models

To determine relevant attributes and explore efficient deep learning techniques for forecasting neuropeptides, we conducted training and evaluation of models utilizing eight distinct features combined with eight prevalent deep learning methods (Fig. 3A). The eight features include ASDC, PSAAC, AAC, DPC, AAE, k-mer, OE and feature extracted by ESM-2. The eight deep learning methods are BiLSTM, CNN, RNN, GRU, Transformer, CNN-Attention, BiLSTM-Attention, and CNN-BiLSTM. Notably, the ESM-2 feature encoding achieved the overall best performance, demonstrating the highest accuracy across all deep learning algorithms, followed by ASDC, PSAAC, AAC, DPC, and then AAE, OE, and k-mer in descending order (Fig. 3A). Among these models, the BiLSTM achieved the highest average accuracy of 0.790. The bidirectional structure of BiLSTM facilitates the incorporation of both preceding and subsequent contextual information of amino acids within the sequence [55]. In comparison to alternative deep learning frameworks, BiLSTM exhibits exceptional suitability for handling sequences and encoding the attributes derived from these sequences. Through its gating mechanism, BiLSTM is able to capture complex interactions between amino acids. Therefore, we selected the BiLSTM for the aforementioned feature fitting tasks. This choice was motivated by the complexity of the features involved, particularly the BLOSUM62 features, which reflect the similarity and substitution frequency between amino acids and contain rich semantic information and complex patterns. Multi-scale convolution can extract features at different scales, simultaneously considering the local similarity between amino acids and the global patterns of the entire sequence. This approach enables the capture of multi-granularity information, enhances feature representation, avoids information loss, and improves the overall performance of the model. Therefore, we chose to use multi-scale convolution to process BLOSUM62 features separately.

3.2. Feature combination optimization based on model performance

Given the promising results achieved by ESM-2 in the field of proteins, we adopted the ESM-2 large model as the basis for our feature set, supplementing it with hybrid feature engineering. Additionally, the BLOSUM62 matrix encoding conserved physicochemical properties has also demonstrated consistent discriminative power in various protein classification domains. Hierarchical multi-scale convolutions were enabled by a CNN architecture with varying kernel sizes, which synergized with BLOSUM62-derived features to capture both local and global sequence patterns. Subsequently, we conducted experiments by initializing our model with features extracted by the large model ESM-2 and features derived from multi-scale CNN using the BLOSUM62 matrix. The initial model demonstrated baseline performances: an ACC of 0.912, with Sn, Sp, MCC, and AUROC scores of 0.868, 0.957, 0.828, and 0.964,

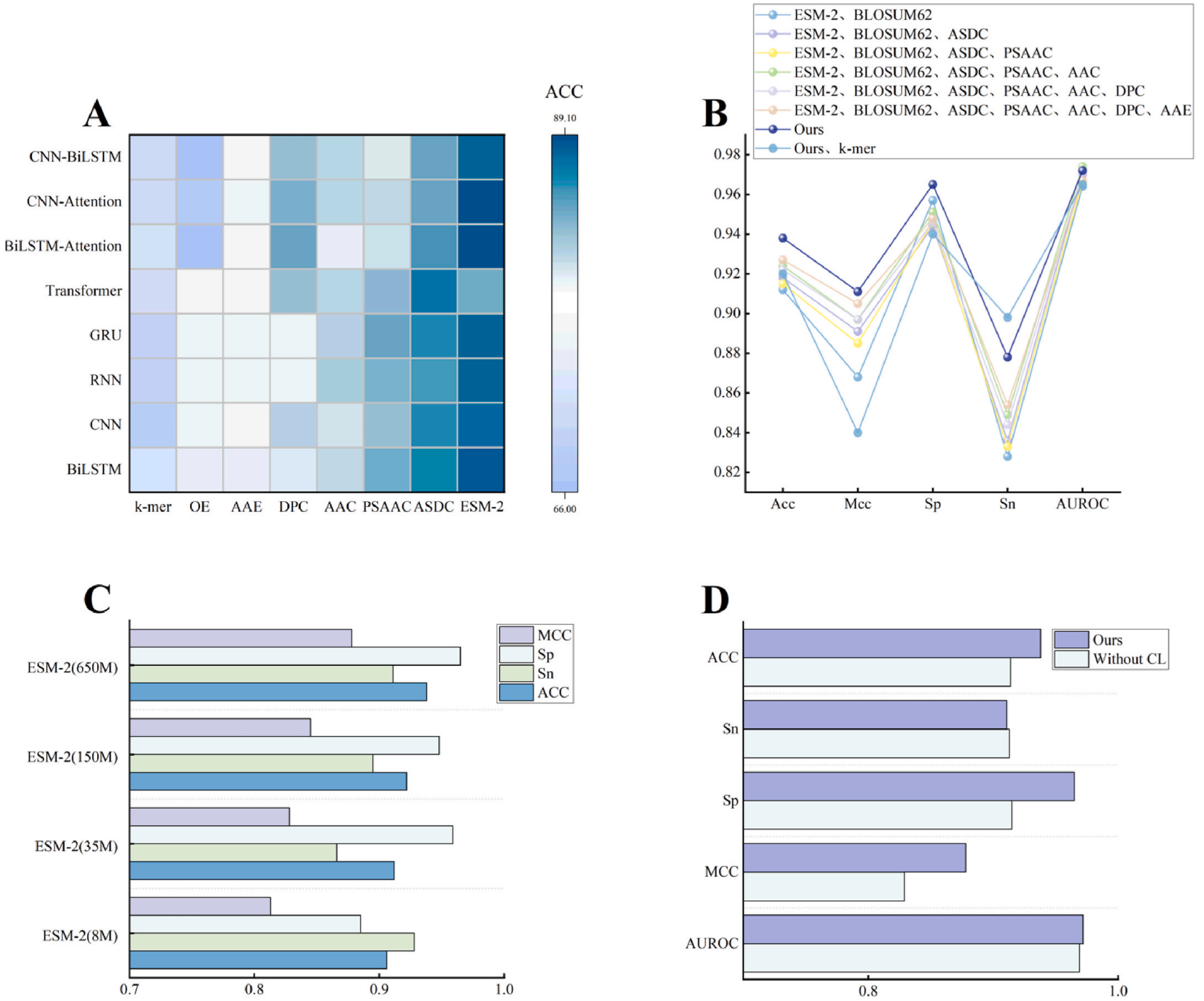

Fig. 3. (A) Heatmap depicting the performance of models utilizing 8 diverse features and 8 various deep learning algorithms. (B) Evaluation of model performance as features are incrementally added. (C) Performance under different parameter sizes of the ESM-2 model. (D) Comparing the performance of models before and after removing contrastive learning.

respectively. By adding ASDC, the improvements in ACC, Sn, MCC, and AUROC are 0.006, 0.023, 0.008, and 0.014, respectively. The PSAAC, AAC, DPC, AAE, and OE were incrementally incorporated. The model's metrics show an overall upward trend. The model's various metrics progressively improved with the addition of each handcrafted feature, reaching a plateau upon the inclusion of k-mer, indicating no further enhancement. However, upon further analysis, we observed that the addition of k-mer features led to a slight decrease in performance, with ACC, Sn, Sp, MCC, and AUROC scores dropping by 0.018, 0.013, 0.025, 0.038, and 0.007, respectively. This suggests that while k-mer features may capture additional sequence information, they may also introduce noise or redundancy that negatively impacts model performance. The final model, which includes all the above features, achieves the exceptional performance across all evaluated metrics, with an ACC, Sn, Sp, MCC, and AUROC scores of 0.938, 0.911, 0.965, 0.878 and 0.972 (Fig. 3B). Ultimately, we utilized ESM-2, BLOSUM62, ASDC, PSAAC, AAC, DPC, AAE, and OE to construct our final model.

3.3. Evaluation of ESM-2 model performance across varying parameter sizes

We performed a systematic scaling analysis on ESM-2 architectures spanning four orders of magnitude. The ESM-2 model with 650 million parameters demonstrates superior performance compared to models with 8 million, 35 million, and 150 million parameters (Fig. 3C). Furthermore, as the number of parameters in the ESM-2 increases, the model's effectiveness gradually improves. A large number of parameters can potentially cause overfitting issues, especially when the volume of data is limited. In our study, we utilized ESM-2 with 3 billion parameters. However, due to overfitting, we did not achieve better results. As the model's parameter count rises, so does the model's representational power and performance. Models of the ESM-2 architecture with varying parameter sizes offer a means to explore the relationship between parameter count and model accuracy. In addition, we compared ESM-2 with ESMC to evaluate their performance in neuropeptide prediction. Contrary to our initial expectations, ESM-2 demonstrated superior performance and stability. The specific architecture and pretraining of ESM-

2 may be particularly well suited for our task. The specific details of the comparison are provided in Supplementary Material S1.

3.4. Ablation studies to evaluate model components

To delve deeper into the impact of the feature decoder and contrastive loss function within our model, as well as various encoding techniques for peptide sequences, we carried out the subsequent three ablation studies.

(i) An analysis contrasting the contrastive learning module with and without the contrast loss function included.
(ii) An evaluation of different encoding approaches applied to the model's input sequences.
(iii) A comparison of different configurations of encoder and decoder in the model.

3.4.1. Ablation of the contrastive learning module

The model equipped with the contrastive loss function outperforms its counterpart without this function, showing superior performance (Fig. 3D). The performance of the model without the contrastive learning module indicated decreases of 0.024, 0.002 in Sn, 0.05 in Sp, 0.049 in MCC, and 0.003 in AUROC, compared to the model with the contrastive learning module. Features derived from large language models such as ESM-2 often encompass extensive semantic and contextual details, whereas manually designed features primarily concentrate on physicochemical properties, positions, and compositional information of peptides. Contrastive learning models have achieved excellent results in bioinformatics field [56]. By employing contrastive learning, the model can exploit the synergistic benefits of these two feature types, capturing diverse aspects of peptide sequence characteristics, ultimately enhancing its predictive capabilities.

3.4.2. Ablation of individual features

Serving as the baseline, the NeuroCL demonstrates superior performance. When the ESM-2 features were removed, the model's performance also declined significantly, with ACC decreasing by 0.065, Sn by 0.101, Sp by 0.029, MCC by 0.126, and AUROC by 0.012. After removing the BLOSUM62 feature, the ACC experienced a drop of 0.033, Sn witnessed a reduction of 0.037, Sp saw a decrease of 0.029, MCC exhibited a decline of 0.066, and AUROC showed a diminution of 0.008, respectively. Additionally, after eliminating the AAC feature, the results indicated decreases of 0.027 in ACC, 0.041 in Sn, 0.012 in Sp, 0.053 in

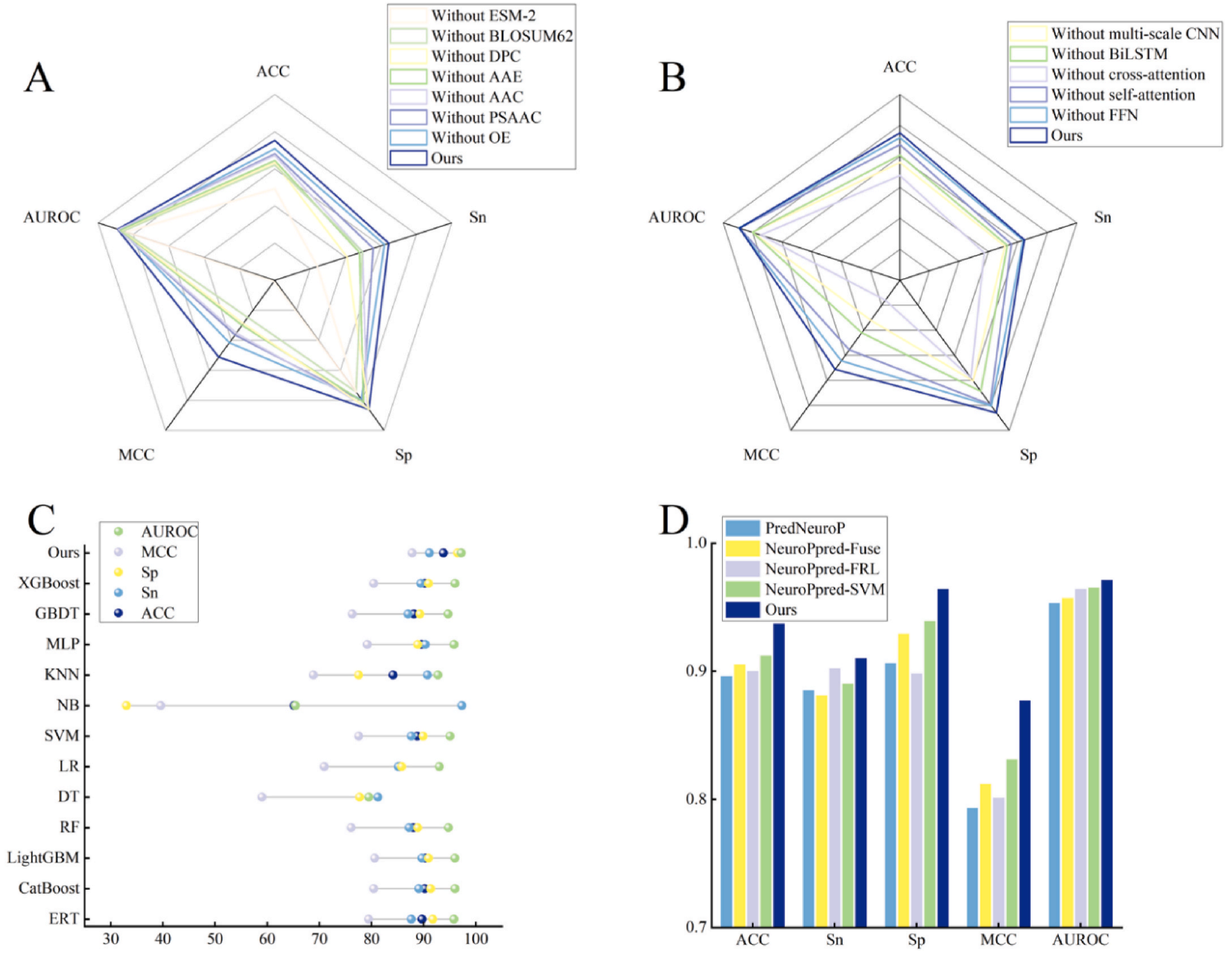

Fig. 4. (A) Analyzing model performance after eliminating each type of manual feature encoding. (B) Performance of models built by deleting every module in the encoder and decoder. (C) Comparing the efficacy of models utilizing diverse machine learning algorithms. (D) Performance comparison of NeuroCL with existing methods based on the independent test set.

MCC, and 0.004 in AUROC. Similarly, removing any other single feature resulted in a decline in the model's performance (Fig. 4A). These findings show that the NeuroCL model's performance decreases when any feature is removed, highlighting the significance of individual attributes in contributing to the model's overall performance. The combination of features enables NeuroCL to capture multidimensional peptide characteristics.

3.4.3. Ablation of each encoder and decoder module

To comprehensively evaluate the contributions of different components in our NeuroCL model, we conducted ablation studies on both the encoder and decoder modules. Starting with the encoder, we observed that removing the multi-scale CNN layers resulted in a noticeable performance drop, with ACC decreasing by 0.048, Sn by 0.033, Sp by 0.064, MCC by 0.098, and AUROC by 0.022. This highlights the importance of multi-scale CNN in capturing both local and global sequence patterns, which are crucial for the model's ability to recognize neuropeptides. Similarly, eliminating the BiLSTM component also led to a decline in performance, with ACC dropping by 0.036, Sn by 0.029, Sp by 0.043, MCC by 0.073, and AUROC by 0.022. This indicates that BiLSTM plays a vital role in modeling the temporal dependencies within the sequence.

Transitioning to the decoder components, we found that the cross-attention mechanism is essential for integrating information effectively. The model without the cross-attention mechanism exhibited a significant decline in performance metrics (Fig. 4B), with a drop in ACC of 0.069, a reduction in Sn of 0.068, a decline in Sp of 0.07, a significant drop in MCC of 0.138, and a reduction in AUROC of 0.036. The results indicate that the manual feature encoding and the information extracted by ESM-2 can be effectively fused through cross-attention, thereby enhancing the expressive power of the features. When the self-attention mechanism was removed, a slight performance decline was observed, with decreases of 0.019 in ACC, 0.022 in Sn, 0.017 in Sp, and 0.039 in MCC. The results suggest that incorporating the self-attention mechanism allows the model to better concentrate on identifying the internal relationships within the sequence, based on these combined features, thus bolstering the feature's representational power. The model without the feed-forward network (FFN) showed a marginal decrease, indicating that the FFN contributes to the model's performance. The FFN performs nonlinear transformations and prevents model degradation, allowing the model to identify more intricate features and patterns. The NeuroCL model achieved optimal performance with all components intact, validating its synergistic architecture for neuropeptide recognition. This demonstrates the rationality and effectiveness of its comprehensive architecture for neuropeptide recognition.

3.5. Comparison with models constructed using various machine learning algorithms

To comprehensively evaluate the performance of our model, we undertook further comparisons against existing machine learning models (Fig. 4C). The results of our model surpassed those of other algorithms, particularly CatBoost, LightGBM, and XGBoost, which, despite demonstrating robust performances, failed to match the level of our model. Compared to machine learning, deep learning exhibits broader application prospects and practical effects in the field of bioinformatics.

3.6. Comparison with other existing methods

To conduct a comprehensive assessment of our model's performance, a standardized test dataset was utilized to NeuroCL against leading-edge models (including NeuroPpred-SVM [21], NeuroPred-FRL [23], NeuroPpred-Fuse [35], and PredNeuroP [17]) under identical evaluation protocols. As illustrated in Fig. 4D, NeuroCL exhibited impressive average metrics, with ACC, Sp, Sn, MCC and AUROC values of 0.938, 0.965, 0.911, 0.878, and 0.976, respectively (Fig. 4D). Notably, NeuroCL demonstrated significant enhancements in ACC, Sp, Sn, MCC, and AUROC compared to the existing optimal models. In addition, to ensure a fair comparison, we have re-evaluated the performance of several existing models using the same 10-fold cross-validation protocol. Specifically, to maintain the authenticity and reliability of our study, we only replicated the experiments for models whose source code was publicly available. NeuroCL exhibited impressive metrics in the 10-fold cross-validation, achieving ACC, Sp, Sn, MCC, and AUROC values of 0.92, 0.948, 0.891, 0.841, and 0.968, respectively (Table 2). Notably, NeuroCL demonstrated significant enhancements in ACC, Sp, Sn, MCC, and AUROC compared to the existing optimal models. Overall, NeuroCL demonstrated superior accuracy in differentiating NPs from non-NPs.

3.7. Visualization of features extracted from each layer of the model

To enhance understanding of our model's ability to identify neuropeptides (NPs), we visualized the outputs from various layers, including input data features, Multi-scale CNN, BiLSTM, cross-attention, self-attention, feed-forward network (FFN), and fully connected layers. To reduce the high-dimensional feature space into two dimensions, we utilized Uniform Manifold Approximation and Projection (UMAP), enabling visualization of the feature distribution. Additionally, we generated kernel density plots for the UMAP-reduced data. These plots display the density distribution of samples along the x and y axes, which correspond to the two dimensions of the reduced feature space. The classification performance of the model can be assessed by the separation between these two curves: the clearer the separation between the curves, the better the model's ability to distinguish between neuropeptides and non-neuropeptides. It is observable that as the neural network layers progressively deepen, only a minority of samples fail to be effectively separated, with positive and negative samples gradually clustering into distinct classes (Fig. 5). Consequently, this demonstrates that NeuroCL is adept at learning representative features that distinguish neuropeptides from non-neuropeptides, highlighting the model's high discriminative power in neuropeptide identification.

4. Webserver construction

To cater to user needs, this framework has been engineered with a user-centric and intuitive interface. Once users access the platform, they simply need to upload a FASTA file containing the sequences. The server will automatically handle the entire prediction process, eliminating the need for users to perform complex calculations or operations. The predicted neuropeptides are presented in a clear and comprehensible format, ensuring that the results are easily understandable. After the prediction is completed, users have the option to download the prediction result file. The web server is freely accessible at http://www.bioai-lab.com/NeuroCL.

5. Conclusion

This research introduces a novel approach, named NeuroCL, designed to predict neuropeptides. The proposed model incorporates the pre-trained language model (ESM-2) alongside seven established feature encoding approaches, including OE, AAC, DPC, PSAAC, ASDC, AAE, and BLOSUM62. It employs cross-attention mechanisms and self-attention mechanisms to deeply extract and fuse features, which are then

Table 2
Performance comparison of NeuroCL with existing methods based on the 10-fold cross-validation results.

Models	ACC	Sn	Sp	MCC	AUROC
PredNeuroP	0.897	0.886	0.907	0.794	0.954
NeuroPpred-Fuse	0.906	0.882	0.930	0.813	0.958
NeuroPpred-SVM	0.913	**0.891**	0.940	0.832	0.966
Ours	**0.920**	**0.891**	**0.948**	**0.841**	**0.968**

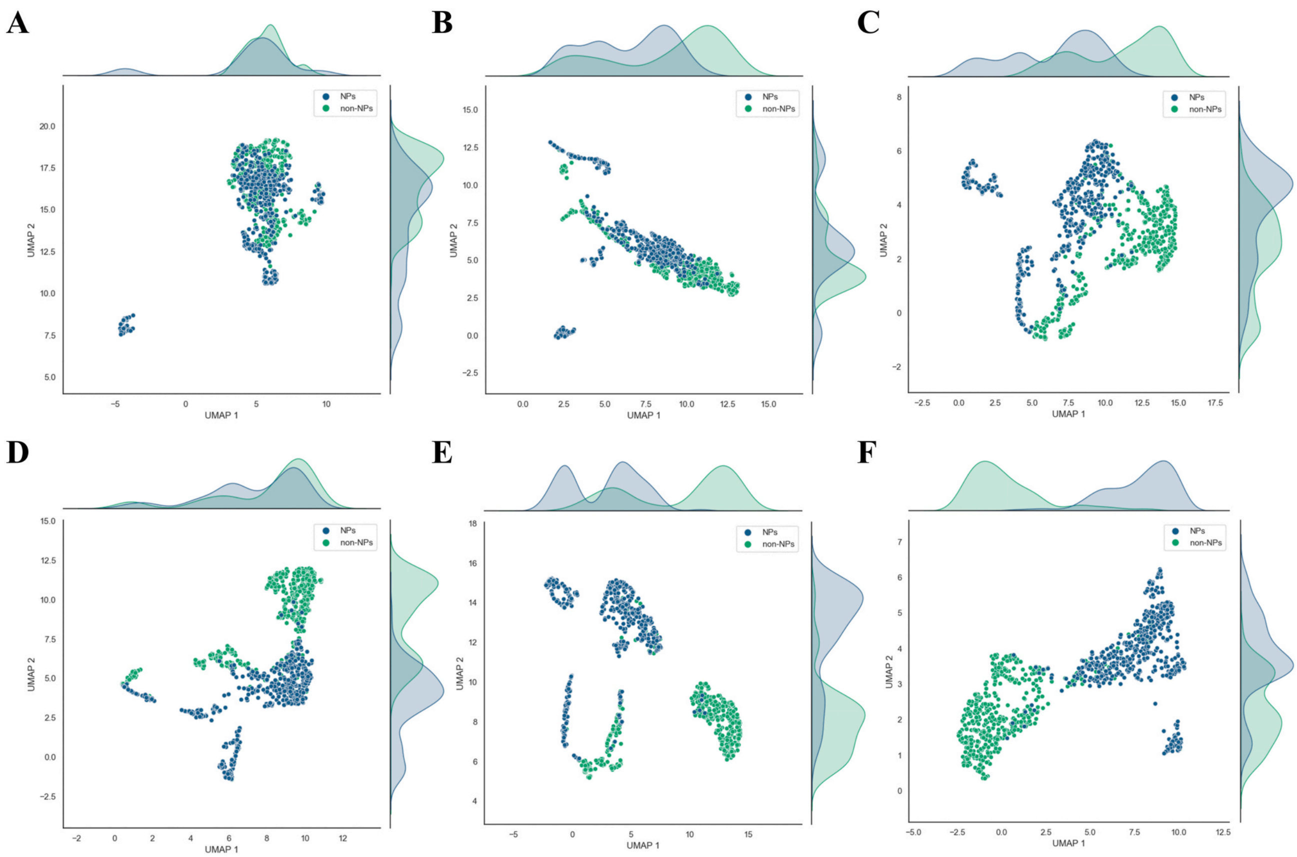

Fig. 5. Visual representation from each layer of the model. (A) Visualization of input layer features. (B) Visualization of Multi-scale CNN and BiLSTM layer features. (C) Visualization of cross-attention layer features. (D) Visualization of self-attention layer features. (E) Visualization of feed-forward network (FFN) layer features. (F) Visualization of output layer features.

transformed into final representations using a feed-forward network. Subsequently, the model optimizes the relationship between features extracted by the large model and traditional features through contrastive learning, and the results are fed into a multilayer perceptron (MLP) for classification. Independent testing results demonstrate that the NeuroCL demonstrates remarkable superiority over current leading models, highlighting its outstanding performance in predicting neuropeptides.

The NeuroCL's innovation lies in several key aspects. By combining the large language model (ESM-2) with conventional feature encoding techniques, it captures both extensive contextual and fine-grained local sequence features, enabling comprehensive feature representation. The integration of cross-attention mechanisms refines this representation by fostering deep connections between features from the large model and traditional methods. Self-attention mechanisms further enhance the model's ability to discern and prioritize crucial information within sequences. Additionally, contrastive learning techniques improve inter-class distinctiveness and intra-class coherence.

Despite its excellent performance in neuropeptide prediction, NeuroCL has limitations. Firstly, while it integrates multiple feature encoding approaches, it may not fully capture all biological determinants influencing neuropeptide functionalities, such as peptide structures or dynamic intermolecular interactions. Secondly, the study relies mainly on computational experiments, lacking the validation of biological experimental corroboration, which may affect the model's credibility and practical applicability. To address these, future research could incorporate a wider range of feature encoding methods based on structural or dynamic information and validate the model's predictions with empirical biological evidence.

CRediT authorship contribution statement

Jian Liu: Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Aoyun Geng:** Writing – review & editing, Methodology, Investigation, Formal analysis, Data curation. **Feifei Cui:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Junlin Xu:** Project administration, Methodology, Investigation, Formal analysis, Data curation. **Yajie Meng:** Supervision, Project administration, Formal analysis, Data curation, Conceptualization. **Leyi Wei:** Formal analysis, Data curation, Conceptualization. **Qingchen Zhang:** Methodology, Investigation, Formal analysis, Conceptualization. **Quan Zou:** Methodology, Formal analysis, Data curation, Conceptualization. **Zilong Zhang:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Generative AI tools ChatGPT in order to improve the readability and language of the manuscript. The authors reviewed and verified all AI-assisted translations to ensure accuracy and consistency with the original content and take full responsibility for the content of the published article.

Fundings

The work is supported by the National Natural Science Foundation of

China (No. 62450002).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ab.2025.115920.

Data availability

Data will be made available on request.The datasets analyzed in the current study are available at https://github.com/niuniunew1203/NeuroCL.

References

[1] C. Lila, R. Delia, D. Daniela, S. Julieta, T. Juan, C. Carla, T.N. Scimonelli, L. Mercedes, Neuropeptides and microglial activation in inflammation, pain, and neurodegenerative diseases, Mediat. Inflamm. 2017 (2017) 5048616.

[2] A.L. Tyburski, L. Cheng, S. Assari, K. Darvish, M.B. Elliott, Frequent mild head injury promotes trigeminal sensitivity concomitant with microglial proliferation, astrocytosis, and increased neuropeptide levels in the trigeminal pain system, J. Headache Pain 18 (2017) 1–12.

[3] C. Weigang, K. Chan-Hee, G. Hye-Jin, E. Michaela, C.G. Zampronio, A.M. Jones, P. N. Gyu, M.R. Elphick, Biochemical, anatomical, and pharmacological characterization of calcitonin-type neuropeptides in starfish: discovery of an ancient role as muscle relaxants, Front. Neurosci. 12 (2018) 382.

[4] V. Kormos, B. Gaszner, Role of neuropeptides in anxiety, stress, and depression: from animals to humans, Neuropeptides 47 (6) (2013) 401–419.

[5] H.C. Mendel, Q. Kaas, M. Muttenthaler, Neuropeptide signalling systems–an underexplored target for venom drug discovery, Biochem. Pharmacol. 181 (2020) 114129.

[6] M. Svensson, K. Sköld, P. Svenningsson, P.E. Andren, Peptidomics-based discovery of novel neuropeptides, J. Proteome Res. 2 (2) (2003) 213–219.

[7] S. Van Bael, J. Watteyne, K. Boonen, W. De Haes, G. Menschaert, N. Ringstad, H. R. Horvitz, L. Schoofs, S.J. Husson, L. Temmerman, Mass spectrometric evidence for neuropeptide-amidating enzymes in Caenorhabditis elegans, J. Biol. Chem. 293 (16) (2018) 6052–6063.

[8] T. Wang, J. Geng, X. Zeng, R. Han, Y.E. Huh, J. Peng, Exploring causal effects of sarcopenia on risk and progression of Parkinson disease by Mendelian randomization, npj Parkinson's Dis. 10 (1) (2024) 164.

[9] J. Martorell-Marugan, M. Chierici, S. Bandres-Ciga, G. Jurman, P. Carmona-Saez, Machine learning applications in the study of Parkinson's disease: a systematic review, Curr. Bioinf. 18 (7) (2023) 576–586.

[10] V. Vapnik, The Nature of Statistical Learning Theory, Springer science & business media, 2013.

[11] P. Agrawal, S. Kumar, A. Singh, G.P. Raghava, I.K. Singh, NeuroPIpred: a tool to predict, design and scan insect neuropeptides, Sci. Rep. 9 (1) (2019) 5129.

[12] X. Qiang, C. Zhou, X. Ye, P-f Du, R. Su, L. Wei, CPPred-FL: a sequence-based predictor for large-scale identification of cell-penetrating peptides by feature representation learning, Briefings Bioinf. 21 (1) (2020) 11–23.

[13] A. Couvineau, S. Dayot, P. Nicole, V. Gratio, V. Rebours, A. Couvelard, T. Voisin, The anti-tumoral properties of orexin/hypocretin hypothalamic neuropeptides: an unexpected therapeutic role, Front. Endocrinol. 9 (2018) 573.

[14] F.Y. Che, R. Biswas, L.D. Fricker, Relative quantitation of peptides in wild-type and Cpefat/fat mouse pituitary using stable isotopic tags and mass spectrometry, J. Mass Spectrom. 40 (2) (2005) 227–237.

[15] H. Zeng, Y. Qin, E. Du, Q. Wei, Y. Li, D. Huang, G. Wang, J.A. Veenstra, S. Li, N. Li, Genomics-and peptidomics-based discovery of conserved and novel neuropeptides in the American cockroach, J. Proteome Res. 20 (2) (2020) 1217–1228.

[16] J.G. Greener, S.M. Kandathil, L. Moffat, D.T. Jones, A guide to machine learning for biologists, Nat. Rev. Mol. Cell Biol. 23 (1) (2022) 40–55.

[17] Y. Bin, W. Zhang, W. Tang, R. Dai, M. Li, Q. Zhu, J. Xia, Prediction of neuropeptides from sequence information using ensemble classifier and hybrid features, J. Proteome Res. 19 (9) (2020) 3732–3740.

[18] J. Kang, Y. Fang, P. Yao, N. Li, Q. Tang, J. Huang, NeuroPP: a tool for the prediction of neuropeptide precursors based on optimal sequence composition, Interdiscipl. Sci. Comput. Life Sci. 11 (2019) 108–114.

[19] S. Karsenty, N. Rappoport, D. Ofer, A. Zair, M. Linial, NeuroPID: a classifier of neuropeptide precursors, Nucleic Acids Res. 42 (W1) (2014) W182–W186.

[20] P.K. Meher, S. Hati, T.K. Sahu, U. Pradhan, A. Gupta, S.N. Rath, SVM-root: identification of root-associated proteins in plants by employing the support vector machine with sequence-derived features, Curr. Bioinf. 19 (1) (2024) 91–102.

[21] Y. Liu, S. Wang, X. Li, Y. Liu, X. Zhu, NeuroPpred-SVM: a new model for predicting neuropeptides based on embeddings of BERT, J. Proteome Res. 22 (3) (2023) 718–728.

[22] Y. Wang, M. Wang, S. Yin, R. Jang, J. Wang, Z. Xue, T. Xu, NeuroPep: a comprehensive resource of neuropeptides, Database 2015 (2015) bav038.

[23] M.M. Hasan, M.A. Alam, W. Shoombuatong, H.-W. Deng, B. Manavalan, H. Kurata, NeuroPred-FRL: an interpretable prediction model for identifying neuropeptide using feature representation learning, Briefings Bioinf. 22 (6) (2021).

[24] S. Gao, Y. Jia, F. Cui, J. Xu, Y. Meng, L. Wei, Q. Zhang, Q. Zou, Z. Zhang, PLPTP: a motif-based interpretable deep learning framework based on protein language models for peptide toxicity prediction, J. Mol. Biol. 437 (12) (2025) 169115.

[25] Q. Yuan, G. Gou, Y. Zhu, Y. Wang, MMCo: using multimodal deep learning to detect malicious traffic with noisy labels, Front. Comput. Sci. 18 (1) (2024) 181809.

[26] H. Zhang, K. Zhang, Z. Li, J. Li, Y. Li, Y. Zhao, Y. Zhu, F. Liu, G. Li, Z. Jin, Deep learning for code generation: a survey, Sci. China Inf. Sci. 67 (9) (2024) 191101.

[27] J. Lv, A. Geng, Z. Pan, L. Wei, Q. Zou, Z. Zhang, F. Cui, iBitter-GRE: a novel stacked bitter peptide predictor with ESM-2 and multi-view features, J. Mol. Biol. 437 (8) (2025) 169005.

[28] S. Gao, Y. Xia, X. Li, F. Cui, Q. Zhang, Q. Zou, Z. Zhang, ACP-ESM2: enhancing anticancer peptide prediction with pre-trained protein language models, IEEE Trans. Comput. Biol. Bioinformat. (2025) 1–14.

[29] X. Zhang, J. Wu, J. Baeza, K. Gu, Y. Zheng, S. Chen, Z. Zhou, DeepTAP: an RNN-based method of TAP-binding peptide prediction in the selection of tumor neoantigens, Comput. Biol. Med. 164 (2023) 107247.

[30] W. Xing, J. Zhang, C. Li, Y. Huo, G. Dong, iAMP-Attenpred: a novel antimicrobial peptide predictor based on BERT feature extraction method and CNN-BiLSTM-Attention combination model, Briefings Bioinf. 25 (1) (2024) bbad443.

[31] F. Barneh, A. Nazarian, R.M. Nadoshan, K.P. Bagheri, A novel in silico filtration method for discovery of encrypted antimicrobial peptides, Curr. Bioinf. 19 (5) (2024) 502–512.

[32] H. Lv, F.-Y. Dao, H. Zulfiqar, H. Lin, DeepIPs: comprehensive assessment and computational identification of phosphorylation sites of SARS-CoV-2 infection using a deep learning-based approach, Briefings Bioinf. 22 (6) (2021) bbab244.

[33] V. Singh, S. Shrivastava, S. Kumar Singh, A. Kumar, S. Saxena, Accelerating the discovery of antifungal peptides using deep temporal convolutional networks, Briefings Bioinf. 23 (2) (2022) bbac008.

[34] A.M. Hayajneh, F. Alasali, A. Salama, W. Holderbaum, Intelligent solar forecasts: modern machine learning models & TinyML role for improved solar energy yield predictions, IEEE Access (2024).

[35] M. Jiang, B. Zhao, S. Luo, Q. Wang, Y. Chu, T. Chen, X. Mao, Y. Liu, Y. Wang, X. Jiang, NeuroPpred-Fuse: an interpretable stacking model for prediction of neuropeptides by fusing sequence information and feature selection methods, Briefings Bioinf. 22 (6) (2021) bbab310.

[36] S. Jiao, X. Ye, C. Ao, T. Sakurai, Q. Zou, L. Xu, Adaptive learning embedding features to improve the predictive performance of SARS-CoV-2 phosphorylation sites, Bioinformatics 39 (11) (2023) btad627.

[37] W. Zhang, J. Liu, M. Zhao, Q. Li, Predicting linear B-cell epitopes by using sequence-derived structural and physicochemical features, Int. J. Data Min. Bioinf. 6 (5) (2012) 557–569.

[38] S. Chen, Y. Liao, J. Zhao, Y. Bin, C. Zheng, PACVP: prediction of anti-coronavirus peptides using a stacking learning strategy with effective feature representation, IEEE ACM Trans. Comput. Biol. Bioinf 20 (5) (2023) 3106–3116.

[39] Y. Zhao, S. Zhang, Y. Liang, HemoFuse: multi-feature fusion based on multi-head cross-attention for identification of hemolytic peptides, Sci. Rep. 14 (1) (2024) 22518.

[40] N. Periwal, P. Arora, A. Thakur, L. Agrawal, Y. Goyal, A.S. Rathore, H.S. Anand, B. Kaur, V. Sood, Antiprotozoal peptide prediction using machine learning with effective feature selection techniques, Heliyon 10 (16) (2024) e103983.

[41] F. Li, X. Guo, D. Xiang, M.E. Pitt, A. Bainomugisa, L.J. Coin, Computational analysis and prediction of PE_PGRS proteins using machine learning, Comput. Struct. Biotechnol. J. 20 (2022) 662–674.

[42] A. Geng, Z. Luo, A. Li, Z. Zhang, Q. Zou, L. Wei, F. Cui, ACP-CLB: an anticancer peptide prediction model based on multichannel discriminative processing and integration of large pretrained protein language models, J. Chem. Inf. Model. 65 (2025).

[43] L. Wang, C. Huang, M. Wang, Z. Xue, Y. Wang, NeuroPred-PLM: an interpretable and robust model for neuropeptide prediction by protein language model, Briefings Bioinf. 24 (2) (2023) bbad077.

[44] X. Wang, B. Zhou, H. Fang, X. Chen, Q. Zhao, K. Xu, Learning to group and label fine-grained shape components, ACM Trans. Graph. 37 (6) (2018) 1–14.

[45] J. Yuan, Z. Wang, Z. Pan, A. Li, Z. Zhang, F. Cui, DPNN-ac4C: a dual-path neural network with self-attention mechanism for identification of N4-acetylcytidine (ac4C) in mRNA, Bioinformatics 40 (11) (2024) btae625.

[46] P. Qu, L. Zhang, Uncertainty-based multi-objective optimization in twin tunnel design considering fluid-solid coupling, Reliab. Eng. Syst. Saf. 253 (2025) 110575.

[47] J. Guan, L. Yao, P. Xie, C.-R. Chung, Y. Huang, Y.-C. Chiang, T.-Y. Lee, A two-stage computational framework for identifying antiviral peptides and their functional types based on contrastive learning and multi-feature fusion strategy, Briefings Bioinf. 25 (3) (2024) bbae208.

[48] C. Xiao, Z. Zhou, J. She, J. Yin, F. Cui, Z. Zhang, PEL-PVP: application of plant vacuolar protein discriminator based on PEFT ESM-2 and bilayer LSTM in an unbalanced dataset, Int. J. Biol. Macromol. 277 (2024) 134317.

[49] X. Fu, Y. Yuan, H. Qiu, H. Suo, Y. Song, A. Li, Y. Zhang, C. Xiao, Y. Li, L. Dou, et al., AGF-PPIS: a protein–protein interaction site predictor based on an attention mechanism and graph convolutional networks, Methods 222 (2024) 142–151.

[50] X. Qin, M. Liu, G. Liu, ResCNNT-fold: combining residual convolutional neural network and Transformer for protein fold recognition from language model embeddings, Comput. Biol. Med. 166 (2023) 107571.

[51] L. Feng, C. Wang, P. Liu, K. Ge, J. Zhang, NCLDR: nearest-neighbor contrastive learning with dual correlation loss for dimensionality reduction, Neurocomputing 594 (2024) 127848.

[52] S. Al-Otaibi, A. Rehman, A. Raza, J. Alyami, T. Saba, CVG-Net: novel transfer learning based deep features for diagnosis of brain tumors using MRI scans, PeerJ Comput. Sci. 10 (2024) e2008.

[53] J. Yue, T. Li, J. Xu, Z. Chen, Y. Li, S. Liang, Z. Liu, Y. Wang, Discovery of anticancer peptides from natural and generated sequences using deep learning, Int. J. Biol. Macromol. (2024) 138880.

[54] Y. Wang, Y. Zhai, Y. Ding, Q. Zou, SBSM-Pro: support bio-sequence machine for proteins, Sci. China Inf. Sci. 67 (11) (2024) 212106.

[55] S. Hochreiter, Long short-term memory, Neural Comput. (MIT-Press 1997).

[56] Y. Yu, M. Gu, H. Guo, Y. Deng, D. Chen, J. Wang, C. Wang, X. Liu, W. Yan, J. Huang, MuCoCP: a priori chemical knowledge-based multimodal contrastive learning pre-trained neural network for the prediction of cyclic peptide membrane penetration ability, Bioinformatics 40 (8) (2024) btae473.