Protein-binding RNA Prediction Based on Integrated Sequence-Structure-Function Pre-training

Lin Gan, Xinyi Wang, Yi Zhou, and Min Zhu

***Abstract*—RNA binding proteins (RBPs) play a crucial role in regulating biological functions through their interactions with specific RNAs, significantly impacting various life processes. High-throughput experiments provide substantial data, facilitating the development of computational predictions. However, current methods struggle to effectively integrate multi-level semantic information and require enhanced predictive accuracy on small-sample datasets. To address these limitations, we propose MTP-RBP, a method that integrates multi-task pre-training with a robust pre-trained encoder. This method not only extracts deep contextual information from RNA sequences but also incorporates structural and functional knowledge for a more comprehensive semantic representation. By enhancing masked language modeling with secondary structure construction and binding function prediction pre-training tasks, MTP-RBP enables better fusion of multi-level features. Experimental results show that MTP-RBP achieves state-of-the-art performance, surpassing baseline and existing RNA language models, particularly on small datasets. The source code of our proposed MTP-RBP can be found in https://github.com/TimmyGan/MTP-RBP.**

***Index Terms*—RNA-protein interaction, deep learning, transformer-based models, Pre-training.**

I. Introduction

RNA binding proteins (RBPs) play a critical role in various fundamental cellular physiological processes [1], encompassing gene expression regulation, post-transcriptional regulation, and protein synthesis [2], [3]. Extensive studies have demonstrated that the disruption of RNA-protein binding can lead to severe cellular dysfunction [4], [5], and their involvement in cancer development and progression has been well-established [6]. Consequently, exploring the RNA-protein interaction holds great potential for gaining novel insights into disease diagnosis and pathogenesis.

In essence, researchers employ high-throughput technologies to identify Protein-binding RNA. Although these experimental approaches are time-consuming and expensive, they yield valuable validated data that significantly promote the development of bioinformatics. In recent years, a multitude of computational methods have emerged for the prediction of Protein-binding RNA. It is commonly acknowledged as a binary classification problem: Each input RNA x is mapped to a label $y \in \{0,1\}$, which indicates whether the RNA contains RBP binding sites. Considering the features employed to describe the input RNA, current computational methods can be classified into two main categories: sequence-based ones and multi-level-semantics-based ones.

First, sequence-based methods aim to infer from RNA sequences solely. The prediction of Protein-binding RNA has traditionally relied on the *specific learning* paradigm, where input sequences are typically represented as feature vectors, and task-specific classification models are trained for prediction. The main contribution of these methods lies in the architectures of classification models to effectively extract critical information from sequences. For example, DeepBind [7] learned the features of RNA using convolutional neural network (CNN). DanQ [8] employs one-hot encoding for RNA and utilizes a CNN-BLSTM architecture to capture both local features and long-range dependencies in RNA sequences. DeepCLIP [9] utilizes a hybrid convolution and BLSTM architecture with WTA-enhancement to improve the model's focus on important RNA sequence features. MCNN [10] employs multiple CNNs to extract RNA vector features from windows of varying lengths, thereby capturing more sequence binding patterns of RBPs. WVDL [11] utilizes one-hot encoded RNA vectors as inputs across three architectures: CNN, CNN-LSTM, and ResNet. The features extracted from these architectures are then combined using a weighted voting method. SA-Net [12] employs the k-mer embedding to represent RNA sequences into dense vectors, and construct a self-attention network resembling a Transformer encoder. However, in the case of learning from a single objective, it's difficult for these specific methods to understand the underlying patterns within RNA sequences. The absence of transferable knowledge limits the prediction performance, especially in the scarcity of training data.

In recent years, the *pre-training - fine-tuning* paradigm has demonstrated a powerful capability superior to task-specific prediction methods. Research spots have shifted to pre-training robust encoders, rather than designing fancy classification models. Inspired by the remarkable success of BERT [13] in natural language processing, researchers have developed pre-trained encoders for biological sequences, thereby benefiting various downstream tasks. For example, DNABERT [14] captures a global and transferable understanding of genomic DNA sequences through masked language modeling. Then BERT-RBP [15] adapts DNABERT for encoding RNA sequences, and achieves remarkable performance on Protein-binding RNA prediction after fine-tuning. RNA-FM [16] performs self-supervised learning on large-scale RNA sequences, thereby revealing latent sequential grammar and evolutionary information. Deservedly, it holds great potential in the prediction of Protein-binding RNA. Despite the remark-

The authors are with the College of Computer Science, Sichuan University, Chengdu 610065, China. E-mail: ganlin228@stu.scu.edu.cn, wangxinyi_scu@163.com, echochou99@163.com, zhumin@scu.edu.cn.

This work was supported by the National Natural Science Foundation of China under Grant 62172289. (Corresponding author: Min Zhu.)

able achievements facilitated by advanced language model techniques, sequence-based methods are inherently insufficient in understanding RNAs comprehensively. There is still room for further improvement in the prediction of Protein-binding RNA by leveraging structural and functional features.

Second, multi-level-semantics-based methods attempt to make full use of multiple sources of information to model RNAs. Existing studies primarily remain in the *specific learning* paradigm, focusing on constructing specific classification models with hybrid architectures. iDeepS [17] utilizes two separate CNNs to learn local features from the encoded RNA sequence and structure, and then applies BLSTM to capture the global information for predictions. EDLMFC [18] integrates both sequence and multi-level structural features, employing a unified CNN-BLSTM architecture for prediction. HLARPBP [19] uses LSTM to extract the correlation relationships between different sites in the RNA sequence, and investigate attention mechanisms in integrating the RNA secondary structure feature. SNB-PSSM [20] utilizes the structural window scheme, incorporating evolutionary information from the spatial neighbors surrounding the target protein residues. aPRBind [21] extends the sequence and structural features by integrating dynamic properties, enabling a more comprehensive feature extraction. MAHyNet [22] employs hybrid CNN and Gated Recurrent Unit (GRU) networks in left-right branches to respectively extract features from RNA sequences and physicochemical properties of RNA bases, and integrates a multi-head attention mechanism for further enhancement. In contrast to sequence-based methods, multi-level-semantics-based methods should hold the potential for comprehensive RNA understanding and greater performance in Protein-binding RNA prediction. However, existing methods are underexploited in addressing the following issues: The specific models struggle in datasets with fewer samples without the support of pre-trained encoders, and hybrid architectures fail to integrate the multi-level semantics of RNAs into unified representations.

In this study, inspired by the within-task pre-training [23], we propose a novel method named MTP-RBP, which introduces Multi-Task Pre-training for Protein-binding RNA prediction. Our pre-trained encoder not only extracts the underlying patterns within RNA sequences, but also has a comprehensive understanding of RNAs with explicit structural and functional knowledge, leading to accurate prediction of Protein-binding RNA.

The contributions of MTP-RBP can be summarised as follows:

(1) We leverage multi-level semantics to describe Protein-binding RNA, including the RNA sequence, RNA secondary structure, and binding function labels.
(2) We introduce a novel multi-task pre-training framework to enable the unified learning of multi-level semantics. Building upon BERT, we augmented the masked language modeling (MLM) with additional tasks, namely secondary structure construction (SSC) and binding function prediction (BFP).
(3) After fine-tuning, our model exhibits state-of-the-art performance superior to relevant baselines and existing RNA language models, especially in datasets with small-scale samples.

II. MATERIALS AND METHODS

In this section, we first describe the dataset used for MTP-RBP. Next, we outline the pre-processing methods for RNA sequences and secondary structures, along with technical details on implementing MTP-RBP.

A. Dataset

In order to assess the MTP-RBP's ability to predict RNA-protein interactions in human CLIP-Seq data, we utilised the publicly available RBP-24 dataset provided by GraphProt [24] (http://www.bioinf.uni-freiburg.de/Software/GraphProt/). The RBP-24 consists of 24 experimental data points for 21 RNA-binding proteins gathered from previous biological research studies. These RBPs exhibit distinct binding characteristics and biological functions, which may contribute to variations across the datasets [25]. In each experiment, positive sites represent subsedquences anchored at the peak center derived from CLIP-seq processed in doRiNA [26], while negative sites denote regions lacking supportive evidence of being binding sites. In addition, the RBP-24 contains independent test sets. Of all the datasets in the entire experiment, 50% were positive samples and the remaining 50% were negative samples. Out of the 24 sub-datasets, 10 have a total sample size of less than 20,000, indicating a relatively limited amount of training data. Consequently, comprehending the effective learning of sequence features from limited data is a crucial aspect of assessing model performance.

To provide a more comprehensive comparison of model performance, we additionally incorporated an eCLIP-seq dataset generated by Pan et al. [27] from the ENCODE3 database [28]. The dataset comprises 154 RBP-specific subsets, each containing up to 60,000 positive RNA sequences bound to the corresponding RBPs, along with an equal number of negative sequences. Each positive sequence spans 101 nucleotides, with the eCLIP-seq peak positioned at its center, while each negative sequence is sampled from non-peak regions of the same reference transcript. Furthermore, we partitioned the dataset into training, validation, and test sets following the strategy proposed by Amada et al. [15].

B. Initial Representation

In this work, we incorporate both sequence and structural feature representations. Specifically, k-mer representation is employed to represent RNA sequence features, while the Byte-Pair Encoding (BPE) [29] is used for the structure construction pre-training task to represent the RNA secondary structure.

1) RNA Sequence

The k-mer representation refers to the extraction of all subsequences of RNA sequences of length k for a better understanding of the characteristics and structure of RNA sequences. k-mer features of the nucleotide composition information of

RNA sequences take into account the local sequence information of RNA sequences, which can characterise the RNA sequences to a certain extent. Specifically, a window of length k nucleotides is used to slide over an RNA sequence in a certain step size, and the subsequence within each window is recorded, which is called k-mer.

Since each position in a k-mer can be a total of four nucleotide types: A, U, G, or C, RNA sequences can generate 4^k-dimensional space. Moreover, considering the filler position N, we treats all subsequences containing N as the same additional type, resulting in a total of $4^k + 1$ unique k-mer subsequence types. For an RNA sequence $s = (s_1, s_2, s_3, ..., s_{l-1}, s_l)$, we employ a sliding window of length k with a step size of 1, resulting in $l - k + 1$ subsequences. Each subsequence is then converted into a one-hot vector with a dimension of $4^k + 1$. Furthermore, the matrix representation of the whole sequence can be obtained by splicing in order, which has the dimension $[l - k + 1,\, 4^k + 1]$.

As the value of k in k-mer increases, the size of the subsequence space grows exponentially, and in order to take into account the computational performance, the length of k-mer generally does not exceed 6 [12]. Here, we set the k-value size to 4, which proved to be optimal in subsequent validations.

2) RNA Structure

RNA secondary structures are used to represent the interaction relationships between contextual bases. In this work, we first predict the RNA secondary structure and then encode the predictions.

For RNA secondary structure prediction, RNAfold [30] was used over other advanced methods such as FocusFold [31], CONTRAFold [32], and LinearFold [33], based on the recommendations by Binet et al. [34]. The predicted structure is represented in a dot-bracket format, where each symbol corresponds to a base in the sequence. Following the principle of base complementary pairing, a secondary structure pairing tree diagram is generated. This diagram is categorized into annotations such as stacking (S), free end (F), joint (J), hairpin (H), internal loop, and multi-loop (M).

For RNA secondary structure encoding, we adapt BPE to model the intricate folding patterns of RNA molecules. Its main principle is to merge frequent adjacent byte pairs in the text to construct a more concise vocabulary. Initially, The vocabulary consists of each RNA secondary structure annotation as a separate morpheme. Next, the algorithm iteratively performs the following operations: first, it counts the frequencies of all adjacent byte pairs; then, it identifies the most frequent byte pair and merges it into a new morpheme; finally, it updates the vocabulary to include this newly merged morpheme. This process repeats until reaching a specified vocabulary size or maximum number of iterations.The merge operation can be described using the following formula:

$$V' = V \cup ab \tag{1}$$

where V is the current vocabulary, ab is the most frequent adjacent byte pair to be merged, and V' is the updated vocabulary after merging.

After multiple merges, the final vocabulary is generated. In this work, we first divide the structural sequence into consecutive subsequences of length k, and then use BPE to represent the RNA secondary structures. In particular, we set the size of the final vocabulary table to 95, which is able to significantly reduce the memory consumption and computational requirements of the model compared to using 4-mer encoded 5^4 dimensions.

C. Model Architectures

The MTP-RBP is divided into two stages: the pre-training stage and the fine-tuning stage, both of which are formed with the Transformer-Encoder architecture [35] as the core. The upstream pre-training performs masked language modeling, binding function prediction and secondary structure construction tasks to represent the functional semantic and structural information of RNA sequences in different contexts; the downstream fine-tuning performs binding function prediction task. The MTP-RBP architecture is shown in Fig.1.

1) Transformer-based Encoding Module

The Transformer-based encoding module consists three key components: the embedding layer, positional encoding, and multi-head attention.

In the embedding layer, for the input RNA matrix $I \in R^{l \times d_{orig.}}$, we use the word embedding dimensionality reduction vector and set the post-embedding dimension to d_{model}. Multiplying the input matrix I with the embedding matrix $M \in R^{d_{orig.} \times d_{model}}$, we obtain the k-mer embedding output matrix $O \in R^{l \times d_{model}}$, which is calculated as.

$$O = \sqrt{d_{model}} IM \tag{2}$$

The introduction of the constant scaling factor $\sqrt{d_{model}}$ is significant as it appropriately enlarges the embedded values to mitigate their numerical impact on subsequent inputs, especially when combined with the positional encoding overlay.

At the stage of positional encoding, sinusoidal and cosinusoidal functions at varying frequencies are employed to assign distinct codes to individual positions within the sequence, enabling the model to differentiate between information at different positions. These formulas are computed as follows:

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{3}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{4}$$

where $pos \in (0,\ 499)$ denotes the position of the subsequence in the whole sequence of k-mer words, and $i \in (0, d_{model}/2 - 1)$ denotes the individual dimensions in which the position of the subsequence is encoded. Subsequently, concatenate the positional encoding vectors in sequence to obtain the positional encoding matrix P. Then, apply a $dropout$ function to the sum of the output matrix O and the positional encoding matrix P.

$$P = [PE_0, PE_1, \ldots, PE_{l-1}]^T \tag{5}$$

$$PE_t = \left[PE_{(t,0)}, PE_{(t,1)}, \ldots, PE_{(t,d_{model}-1)}\right]^T \tag{6}$$

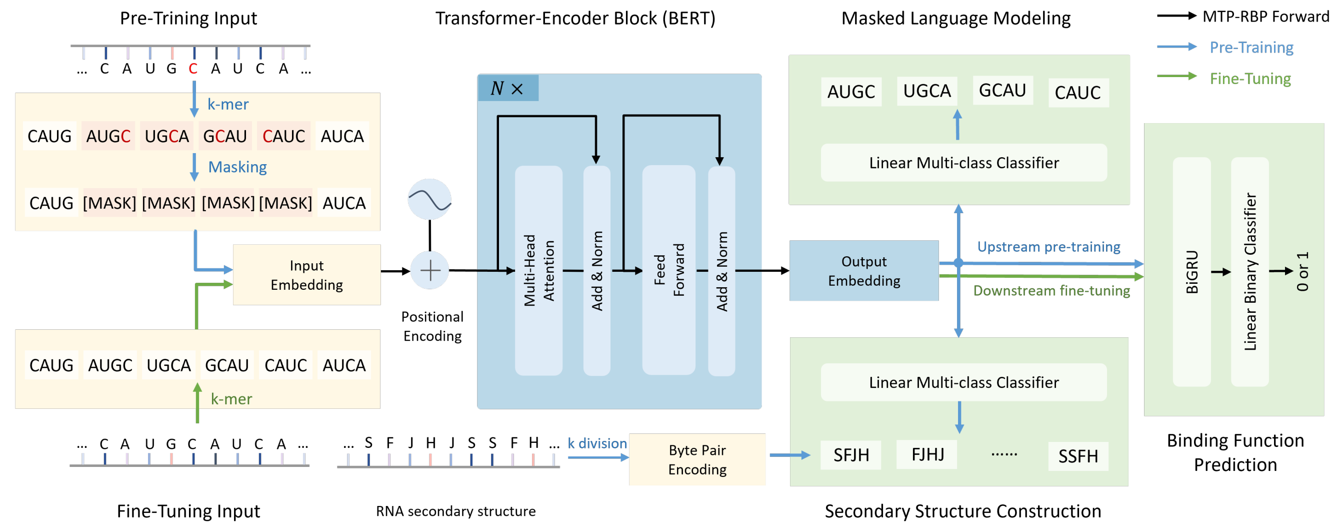

Fig. 1. The network framework flowchart of MTP-RBP. The model comprises Feature Representation, Transformer-Encoder Block, and three task modules, namely Mask Language Modeling, Binding Function Prediction, and Secondary Structure Construction. First, the RNA sequences are encoded by k-mer representation, and BPE was performed on the RNA secondary structure. Subsequently, the Transformer-Encoder is pre-trained in accordance with the three aforementioned task modules, thereby enabling the model to obtain a more comprehensive sequence-structure-function feature representation. Finally, the fine-tuned stage shares some of the parameters corresponding to the pre-trained stage for initialization and performs the Binding Function Prediction task to predict whether the RNA sequence can bind to the protein.

$$X = dropout\left(P + O\right) \tag{7}$$

where the matrix $X \in R^{l \times d_{model}}$ represents the input sequence matrix after adding the position encoding, and the $dropout$ function takes p_{drop} as the random inactivation probability.

At the multi-head attention section, it consists of N identical Transformer-Encoder layers, where the output of each layer will be used as input to the next layer. Each layer contains multi-head attention sublayer and position feed forward network sublayer. After completing the computations in the sublayers, a dropout function is applied to the output, and residual connections with layer normalization are employed.

$$H\left(x\right) = LayerNorm\left(x + dropout\left(Sublayer\left(x\right)\right)\right) \tag{8}$$

where $Sublayer\left(x\right)$ denotes the function implemented by the sublayer. To simplify the residual linkage computation, each sublayer generates outputs of dimension d_{model}.

The multi-head attention allows the Transformer-Encoder to focus on the information from different subspaces, which helps our model to learn more RNA sequence features. In RNA sequences, padding positions do not carry meaningful information and should not be assigned attention weights. Therefore, padding positions should be masked, meaning the attention weights are set to 0 during computation, ensuring that the model does not attend to the padded parts of the sequence.

$$MultiH\left(Q, K, V\right) = Concat\left(head_1, \ldots, head_h\right) W^O \tag{9}$$

$$head_i = Attention\left(XW_i^Q, XW_i^K, XW_i^V\right) \tag{10}$$

$$Attention\left(\widetilde{Q},\ \widetilde{K},\ \widetilde{V}\right) = softmax\left(\frac{\widetilde{Q}\widetilde{K}^T}{\sqrt{d_k}}\right)\widetilde{V} \tag{11}$$

where W is a linear propagation parameter and d_k denotes the output dimension of each head computed by attention, which is set to d_{model}/h; h denotes the number of heads with multi-head attention.

The position feed-forward network sublayer consists of two linear transformations with a $GELU$ activation function in the middle.

$$FFN\left(x\right) = GELU\left(xW_1 + b_1\right) W_2 + b_2 \tag{12}$$

$$GELU(x) = x \cdot \varphi(x) \tag{13}$$

where $W_1 \in R^{d_{model} \times d_{ff}}$ and $W_2 \in R^{d_{ff} \times d_{model}}$ are the transformation parameters, d_{ff} denotes the dimensionality after the first linear transformation, b_1 and b_2 denote the bias terms added in the two linear changes, respectively. $\varphi(x)$ represents the cumulative distribution function (CDF) of the standard normal distribution, which is given by:

$$\varphi(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{14}$$

where erf(x) is the error function.

2) *Classification Modules*

Classification modules correspond to the learning of sequence-structure-function multiple semantics features, respectively, where sequence and structure features use liner multi-class classifier, and functional features use BiGRU-based binary classifier.

a) *Liner Multi-class Classifier for Sequence*

The liner multi-class classifier for sequence primarily predicts the k-mer subsequences of RNA that are masked. It takes the output matrix $A \in R^{l \times d_{model}}$ from the Transformer-Encoder, and selects the masked input sequence positions in the matrix to form the output vector for the multi-classifier input matrix $B \in R^{l_m \times d_{model}}$. The input matrix B is linearly transformed once in equal dimensions with $GELU$ as the activation function, normalized by one layer, and then linearly transformed again with $softmax$ as the activation function, and projected to the dictionary dimension to predict the k-mer subsequences masked by the [MASK] token.

$$C = GELU\left(BW_B + b_B\right) \tag{15}$$

$$P = softmax\left(LayerNorm(C)W_C + b_C\right) \tag{16}$$

where the linear variation matrix parameters $W_B \in R^{d_{model} \times d_{model}}$ and $W_C \in R^{d_{model} \times 258}$ are randomly initialized and subsequently optimized by the back-propagation algorithm of the model, and b_B and b_C denote the bias terms added to the two linear variations, respectively.

b) Liner Multi-class Classifier for Structure

Unlike the multi-class classifier for sequence, which predicts masked sequences, the multi-classifier for structure is used to predict the entire secondary structure of RNA. It receives the output matrix $A \in R^{l \times d_{model}}$ from the Transformer encoder. Initially, matrix A undergoes a fully connected layer operation with the $GELU$ activation function, followed by layer normalization. Subsequently, another fully connected layer operation is performed using the $softmax$ function, projecting the results into the BPE encoding dimension to predict RNA secondary structure fragments.

$$B = GELU\left(AW_A + b_A\right) \tag{17}$$

$$P = softmax\left(LayerNorm(B)W_B + b_B\right) \tag{18}$$

where W is the linear variation matrix parameter; b_A and b_B denote the bias terms in the two linear variations, respectively.

c) BiGRU-based binary Classifier for Function

The binary classifier is used to determine whether RNA has binding functionality. It further captures long-term dependencies and contextual information in the sequence data by adding a BiGRU layer to receive the output matrix A of the Transformer-Encoder, and a binary classifier is then used to classify whether the masked sequence has binding site semantics.

The GRU consists of two gating units, reset gate r_t and update gate z_t. It computes the output h_t of the current hidden node by combining the current input x_t and the contained hidden state h_{t-1} inherited from the previous node.

$$r_t = \sigma\left(w_r \cdot [h_{t-1}, x_t]\right) \tag{19}$$

$$z_t = \sigma\left(w_z \cdot [h_{t-1}, x_t]\right) \tag{20}$$

where w_r and w_z are the weights of the reset and update gates. σ is the logistic sigmoid function. Afterwards, the reset data $\acute{h}_{t-1}$ spliced with x_t is obtained using the reset gate r_t and scaled by a $tanh$ activation function.

$$\tilde{h}_t = \tanh\left(w_{\tilde{h}} \cdot \left[\ \acute{h}_{t-1}, \quad x_t\ \right]\right) \tag{21}$$

$$\acute{h}_{t-1} = h_{t-1} \cdot r_t \tag{22}$$

where h_t denotes the candidate memory content. Finally, the GRU calculates the final hidden state h_t as output.

$$h_t = z_t \cdot \tilde{h}_t + (1 - z_t) \cdot h_{t-1} \tag{23}$$

Further, BiGRU takes into account reverse RNA sequences by combining forward GRUs with reverse GRUs, allowing the model to learn more contextual information and improve classification accuracy.

Subsequently, the output of BiGRU is sent to a binary classifier. The biclassifier does an equal dimensional linear transformation of the output with a $tanh$ activation function. After spreading, the probability p is calculated using a linear transformation with a $sigmoid$ activation function.

D. Details in Pre-training and Fine-tuning

Corresponding to the three semantic information categories of sequence, structure, and function, we have established the following three pre-training tasks:

Task 1: Masked Language Modeling (MLM): The MLM aims to learn the sequence semantic features of RNA from the word level. K-mer subsequences are in different contexts and the corresponding functional semantics are somewhat different, the model predicts the masked word by the context to learn the functional semantics of the word in each context. First, 15% of the k-mer subsequences are selected to be replaced with [MASK] tokens, and to increase the difficulty of prediction, we mask the consecutive k subsequences at a time. Second, the model predicts the corresponding subsequence of [MASK] to train the model parameters according to the context. The sparse categorical cross-entropy loss function is employed for the MLM task, ensuring the model effectively captures and learns these nuanced semantic features.

$$L_{MLM} = -\frac{1}{M}\sum_{i=1}^{M} \log p_{i,y_i} \tag{24}$$

Task 2: Binding Function Prediction (BFP): The BFP aims to determine whether an RNA contains binding site functional semantics from the sentence level. In this context, a ”sentence” in natural language is analogous to the entire RNA sequence. This task involves adding a BiGRU-based binary classifier that receives semantic embedding matrices to classify whether the masked sequence possesses binding site semantics. Sequences classified as 0 do not have binding site semantics and do not interact; sequences classified as 1 have binding site semantics and interact with RNA-binding proteins. The mean square error loss function is used for the BFP task, ensuring precise classification by minimizing the error between predicted and actual classifications.

$$L_{BFP} = \frac{1}{M}\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 \tag{25}$$

Task 3: Secondary Structure Construction (SSC): The SSC aims to extract RNA features from the structural hierarchy. The encoder is capable of containing partial RNA structural information following the completion of the MLM pre-training task [16]. However, training only word-level and sentence-level tasks during the pre-training stage is insufficient for fully representing structural features. Therefore, in order to further highlight the informative representation of RNA secondary structure, we set the task of constructing the whole secondary structure. By mapping predicted structures to the en coding dimensions of BPE, the model comprehensively captures the features of RNA structures. The sparse categorical cross-entropy loss function is employed for the SSC task, ensuring accurate and robust representation of the RNA secondary structure.

$$L_{SSC} = -\frac{1}{M}\sum_{i=1}^{M}\log p_{i,y_i} \tag{26}$$

In the pre-training phase, the training data adopted for the MLM task and the SSC task are all positive samples in the training set, which are used to model the semantic information of RNA from sequences and structural hierarchy. The training data used for the BFP task is the entire training set, which is used to learn sequences with semantic information of binding sites from the functional hierarchy.

During pre-training, we employ a step-by-step pre-training approach. First, we execute the MLM task and the BFP task to extract the RNA sequence context and functional information. After that, all three pre-training tasks are executed simultaneously, which enables the pre-trained encoder to characterize the comprehensive features at the three levels of RNA sequence-structure-function. The overall loss function in the pre-training is the sum of the loss functions of the three tasks. The maximum number of iterations of pre-training epoch is set to 50. we use the early stop method to stop the model training early when the loss of the model no longer continues to decline, and save parameters to the checkpoint.

In the fine-tuning phase, the fine-tuned model is initialized by sharing some of the parameters corresponding to the pre-trained model. The model is fine-tuned using the mean square error loss function on each of the sub-datasets, and the fine-tuning process employs mini-batch gradient descent with a batch size of 128 and the Adam optimization function with a learning rate set to 0.001. The details of hyperparameter settings are shown in Table I.

TABLE I
HYPER -PARAMETER S ETTINGS OF MTP-RBP

Hyper-parameters	Setting
k-mer embedding encoding k	4
k-mer embedding dimension d_{model}	16
Transformer-Encoder stacking number N	6
Number of heads in the multi-head attention sublayer h	2
dropout function's dropout rate p_{drop}	0.1

III. RESULT

To evaluate the performance of MTP-RBP, we conducted a series of experiments on the RBP-24 and ENCODE datasets. First, we compared the performance of MTP-RBP with six other methods. Next, we performed ablation studies on the representation strategy, model pre-training strategy, and model encoding architecture. Finally, we evaluated the performance of the functional prediction module during the fine-tuning classification stage and analyzed the performance of BiGRU in comparison to the widely used BLSTM and CNN-BLSTM.

A. Evaluation Metrics

Consistent with the previous evaluation metrics for the Protein-binding RNA prediction problem, we use the area under the ROC curve (AUC) as an evaluation metric to assess the performance of MTP-RBP. Additionally, we employ average precision score (AP), Recall, and Matthews Correlation Coefficient (MCC) to provide a more comprehensive performance evaluation. AUC is computed based on True Positive Rate (TPR) and False Positive Rate (FPR), while AP is calculated from Precision and Recall. The definitions of these metrics are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{27}$$

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

$$TPR = \frac{FP}{FP + TN} \tag{29}$$

$$FPR = \frac{TP}{FP + TN} \tag{30}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{31}$$

where TP, TN, FP and FN represent the number of true-positive, true-negative, false-positive and false-negative samples, respectively. AUC denotes the area enclosed by the ROC curve with the axes using TPR and FPR as the vertical and horizontal axes. AP denotes the area enclosed by the Precision-Recall curve with the axes.

B. Comparison With Advanced Methods

In this section, we first compare MTP-RBP with the leading methods. Next, we compare our approach with RNA pre-trained language model encoding methods.

1) Methods Specific to Protein-binding RNA Prediction

To demonstrate the exceptional performance of our proposed MTP-RBP model, we conduct a comparative study on the RBP-24 and ENCODE datasets against five typical and state-of-the-art methods: iDeepV [36], iDeepS [17], DeepCLIP [9], SA-Net [12], and WVDL [11]. Below is a brief description of each baseline model.

- iDeepV: Using RNA sequences as input, k-mer embeddings learned through the Word2Vec algorithm as feature representations, and CNNs are utilized for classification.
- iDeepS: Using RNA sequences and RNA secondary structures as inputs, one-hot encoding as feature representation, and CNN-BLSTM architecture for classification.
- DeepCLIP: Using RNA sequences as input, one-hot encoding as feature representation, and a WTA-enhanced CNN-BLSTM architecture for classification.

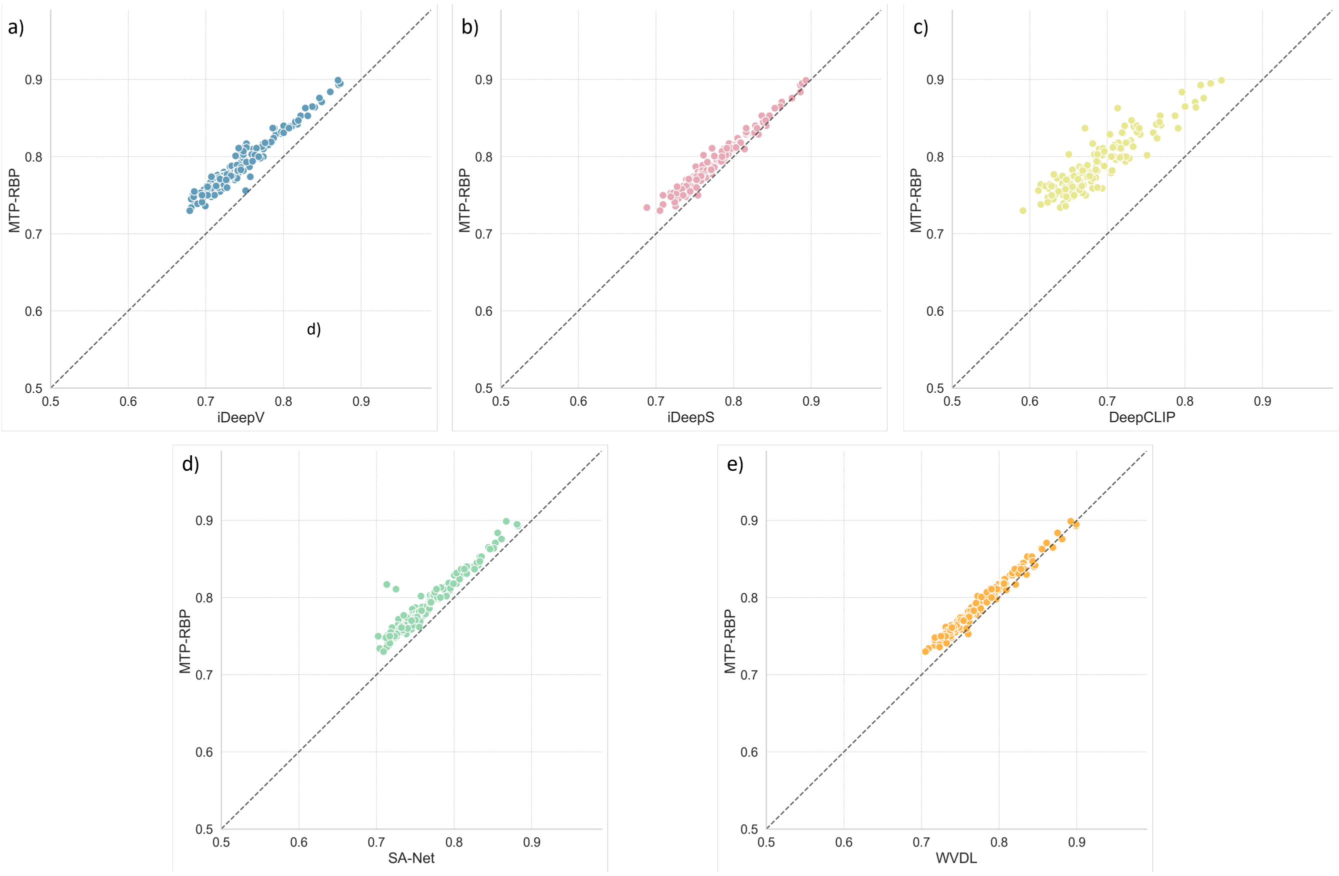

Fig. 2. The AUC of MTP-RBP and other models across ENCODE's 154 RBP datasets. Each dot represents the AUC of MTP-RBP and the corresponding baseline model trained using the same RBP dataset. The dots above the dotted line indicate that the MTP-RBP model performs better than the other models.

- SA-Net: Using RNA sequences as input, k-mer encoding as feature representation, and using Self-attention based neural network for classification.
- WVDL: Using RNA sequences as input, one-hot encoding as feature representation, and using Weighted Voting method to combine multiscale CNN-LSTM and ResNet for classification.

TABLE II
THE PERFORMANCE OF METHODS ACROSS RBP-24 AND ENCODE

Dataset	Model	AUC	AP	Recall	MCC
RBP-24	iDeepV	0.913	0.907	0.863	0.684
	iDeepS	0.933	0.935	0.864	0.756
	DeepCLIP	0.935	0.931	0.870	0.743
	SA-Net	0.945	0.945	0.893	0.767
	WVDL	0.952	0.951	**0.908**	0.801
	MTP-RBP	**0.961**	**0.961**	**0.908**	**0.816**
ENCODE	iDeepV	0.746	0.737	0.688	0.361
	iDeepS	0.773	0.765	0.705	0.404
	DeepCLIP	0.688	0.675	0.639	0.272
	SA-Net	0.764	0.755	0.697	0.391
	WVDL	0.774	0.764	0.708	0.411
	MTP-RBP	**0.790**	**0.782**	**0.730**	**0.434**

Bold: *best results.*

Table II presents a performance comparison of our model with existing methods across RBP-24 and ENCODE datasets. Our model consistently outperforms the baselines, achieving the best overall results across all four metrics on both datasets. Specifically, on RBP-24, MTP-RBP attained the highest AUC (0.961), exceeding iDeepV, iDeepS, DeepCLIP, SA-Net, and WVDL by 4.8%, 2.8%, 2.6%, 1.6%, and 0.9%, respectively. Similarly, on the ENCODE dataset, MTP-RBP achieved the highest AUC (0.790), surpassing iDeepV, iDeepS, DeepCLIP, SA-Net, and WVDL by 4.4%, 1.7%, 10.2%, 2.6%, and 1.6%, respectively. Fig.2 further illustrates that in a one-to-one comparison with each baseline, our model obtained the highest scores in 141 out of 154 RBPs in the ENCODE dataset. Likewise, as shown in Table III, our model outperformed all baselines in 21 out of 24 RBPs in the RBP-24 dataset. These results demonstrate that MTP-RBP consistently maintains superior predictive performance across different datasets, further confirming its robustness and generalizability.

It is worth emphasizing that on the RBP-24 dataset, our model demonstrates significant performance improvements, particularly on sub-datasets with fewer training samples. For example, on the ALKBH5 sub-dataset with 2,410 samples and the C17ORF85 sub-dataset with 3,709 samples, MTP-RBP achieves AUC values of 0.842 and 0.938, respectively, surpassing the best results of other models by 5.8% and 3.6%. This indicates that our model, trained with a specific pre-training task, effectively captures deep contextual representations of RNA sequences, exhibits strong generalization ability, and achieves excellent performance even with limited

TABLE III
THE AUC PERFORMANCE OF MTP-RBP AND OTHER METHODS ACROSS RBP-24

RBP	iDeepV	iDeepS	DeepCLIP	SA-Net	WVDL	MTP-RBP
ALKBH5	0.643	0.773	0.716	0.788	0.784	**0.842**
C17ORF85	0.740	0.865	0.898	0.902	0.886	**0.938**
C22ORF28	0.823	0.842	0.838	0.869	0.874	**0.899**
CAPRIN1	0.824	**0.961**	0.948	0.919	0.948	0.950
Ago2	0.886	0.834	0.859	0.898	0.899	**0.911**
ELAVL1(H)	0.966	0.979	0.981	0.983	0.980	**0.983**
SFRS1	0.905	0.952	0.955	0.960	0.963	**0.973**
HNRNPC	0.979	0.978	0.983	0.981	0.983	**0.985**
TDP43	0.935	0.914	0.905	0.946	0.954	**0.968**
TIA1	0.941	0.941	0.945	0.952	0.954	**0.965**
TIAL1	0.929	0.938	0.943	0.938	0.948	**0.959**
Ago1-4	0.925	0.915	0.918	0.933	0.940	**0.950**
ELAVL1(B)	0.962	0.978	0.982	0.982	**0.984**	**0.984**
ELAVL1(A)	0.973	0.974	0.982	0.977	0.975	**0.984**
EWSR1	0.962	0.972	0.974	0.971	0.981	**0.986**
FUS	0.976	0.983	0.986	0.984	0.991	**0.993**
ELAVL1(C)	0.990	**0.997**	0.995	0.992	0.993	0.994
IGF2BP1-3	0.923	0.906	0.898	0.966	0.970	**0.975**
MOV10	0.896	0.931	0.940	0.940	**0.958**	0.953
PUM2	0.965	0.975	0.969	0.970	0.972	**0.986**
QKI	0.965	0.958	0.975	0.973	0.974	**0.988**
TAF15	0.978	0.975	0.982	0.976	0.982	**0.989**
PTB	0.936	0.930	0.927	0.957	0.950	**0.958**
ZC3H7B	0.883	0.929	0.933	0.928	0.953	**0.954**
Average	0.913	0.933	0.935	0.945	0.952	**0.961**

Bold*: best results.*

training data. Meanwhile, SA-Net performs better than other methods on these two sub-datasets due to its use of the attention mechanism. Although both SA-Net and MTP-RBP utilize RNA sequences as inputs, the integrated pre-training strategy in MTP-RBP enables the model to incorporate comprehensive sequence-structure-function information, thereby enhancing prediction accuracy.

2) *Methods Based on RNA Language Models*

To assess the broad applicability of MTP-RBP in RNA-protein interactions, we compared it with the pre-trained language model encoding method RNA-FM. RNA-FM utilizes only MLM as its pre-training task. In contrast, MTP-RBP incorporates additional pre-training tasks at both structural and functional levels.

TABLE IV
COMPARISON OF ENCODING PERFORMANCE BETWEEN MTP-RBP AND RNA-FM ACROSS RBP-24 AND ENCODE

Dataset	Model	AUC	AP	Recall	MCC
RBP-24	RNA-FM MTP-RBP	0.938 **0.961**	0.938 **0.961**	0.878 **0.908**	0.762 **0.816**
ENCODE	RNA-FM MTP-RBP	0.752 **0.790**	0.745 **0.782**	**0.764** 0.730	0.352 **0.434**

Bold*: best results.*

Table IV presents the four evaluation metrics of the two deep encoding methods across RBP-24 and ENCODE datasets. MTP-RBP consistently outperforms RNA-FM across both datasets in terms of AUC, AP, and MCC. Specifically, on the RBP-24 dataset, MTP-RBP achieves an AUC of 0.961, surpassing RNA-FM by 2.3%, while also demonstrating higher AP and MCC. Furthermore, MTP-RBP attains a recall of 0.908, indicating a higher sensitivity in identifying true binding sites. Similarly, on the ENCODE dataset, MTP-RBP achieves an AUC of 0.790, outperforming RNA-FM by 3.8%. It also exhibits higher AP and MCC, further confirming its robustness in RNA-protein interaction prediction. While RNA-FM achieves a slightly higher recall , this comes at the cost of lower precision, as reflected by the lower MCC score. These results suggest that the incorporation of additional pre-training tasks at both structural and functional levels in MTP-RBP enhances its feature representation capabilities, leading to superior predictive performance compared to RNA-FM.

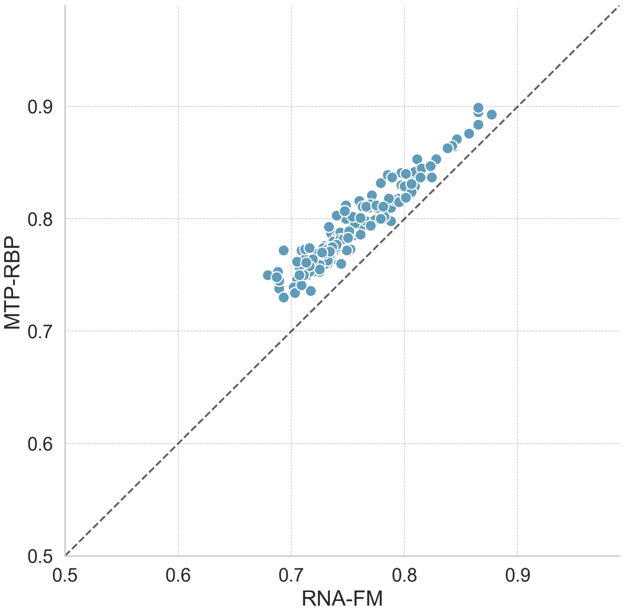

Fig. 3. The AUC of MTP-RBP and RNA-FM over ENCODE's 154 RBP datasets. The dots above the dotted line indicate that the MTP-RBP performs better than the RNA-FM.

Fig. 3 shows that in a one-to-one comparison with RNA-FM, our model achieves the highest scores across all 154 RBPs in the ENCODE dataset. Similarly, Table V presents the AUC and AP of the two deep encoding methods on the RBP-24 dataset. MTP-RBP obtained the highest AUC in 22 out of 24 RBPs and the highest AP in 20 out of 24 RBPs, demonstrating that our BFP and SSC pre-training tasks effectively capture the structural and functional features of RNAs, providing a more comprehensive representation of the sequence code. In the sub-datasets with the smallest sample sizes, ALKBH5 and C17ORF85, MTP-RBP outperforms RNA-FM by 5.2% and 3.7%, respectively. These results highlight the strong applicability of our pre-training and encoding methods for small datasets.

C. *Ablation Experiments*

To assess the efficacy of individual modules within MTP-RBP, in this section, we conduct ablation experiments across three dimensions: pre-training strategies, representation strategies, and model architectures.

TABLE V
COMPARISON OF ENCODING PERFORMANCE ON RBP-24

RBP	RNA-FM		MTP-RBP	
	AUC	AP	AUC	AP
ALKBH5	0.790	0.791	**0.842**	**0.852**
C17ORF85	0.901	0.898	**0.938**	**0.938**
C22ORF28	0.870	0.875	**0.899**	**0.886**
CAPRIN1	**0.951**	0.944	0.950	**0.954**
Ago2	0.850	0.854	**0.911**	**0.915**
ELAVL1(H)	**0.984**	**0.985**	0.983	0.984
SFRS1	0.957	0.954	**0.973**	**0.973**
HNRNPC	0.984	**0.981**	**0.985**	0.978
TDP43	0.922	0.930	**0.968**	**0.969**
TIA1	0.949	0.946	**0.965**	**0.968**
TIAL1	0.936	0.933	**0.959**	**0.959**
Ago1-4	0.925	0.926	**0.950**	**0.950**
ELAVL1(B)	0.983	**0.985**	**0.984**	0.984
ELAVL1(A)	0.973	0.966	**0.984**	**0.982**
EWSR1	0.975	0.975	**0.986**	**0.986**
FUS	0.986	0.988	**0.993**	**0.994**
ELAVL1(C)	**0.994**	**0.995**	**0.994**	0.993
IGF2BP1-3	0.889	0.889	**0.975**	**0.975**
MOV10	0.940	0.938	**0.953**	**0.958**
PUM2	0.964	0.972	**0.986**	**0.988**
QKI	0.968	0.975	**0.988**	**0.990**
TAF15	0.976	0.972	**0.989**	**0.991**
PTB	0.930	0.923	**0.958**	**0.959**
ZC3H7B	0.927	0.911	**0.954**	**0.949**
Average	0.939	0.938	**0.961**	**0.961**

***Bold**: best results.*

1) Pre-training Strategies

To confirm that MTP-RBP achieves the best encoding performance for RNA sequences when all three pre-training tasks are executed, we conducted pre-training strategy ablation experiments as follows: Maintaining consistency in the MTP-RBP model architecture and hyper-parameter settings, we categorized it into four models:

- $\text{Model}_{\text{None}}$: Without pre-training.
- $\text{Model}_{\text{MLM}}$: Pre-trained only with the MLM task.
- Model_{MB}: Pre-trained with MLM and BFP tasks.
- $\text{Model}_{\text{Struct}}$: Extends $\text{Model}_{\text{MLMBFP}}$ by incorporating RNA secondary structure as additional input. This model concatenates encoded sequences with structural embedding features for input into the subsequent prediction module.

Fig.4 shows the AUC on RBP-24 for models with different pre-training strategies. Compared to the model without pre-training, the performance of the model pre-trained only with the MLM task exhibited some improvement, but was significantly lower than the model pre-trained with both strategies. The experimental results illustrate that the model pre-trained only with MLM learns the semantic information of k-mer subsequences at the sequence level. In contrast, the model pre-trained with both strategies incorporates binding site semantic information at the function level, enhancing its ability to capture deeper contextual aspects of RNA sequences. MTP-RBP achieves optimal performance when employing all three pre-training tasks simultaneously. This approach effectively models the contextual representation of sequences while seamlessly integrating RNA structural information. Moreover, MTP-RBP exhibits a significant advantage over $\text{Model}_{\text{Struct}}$, which directly incorporates structural features as inputs, indicating that concurrently performing multiple pre-training tasks facilitates the fusion of sequence-structure-function features.

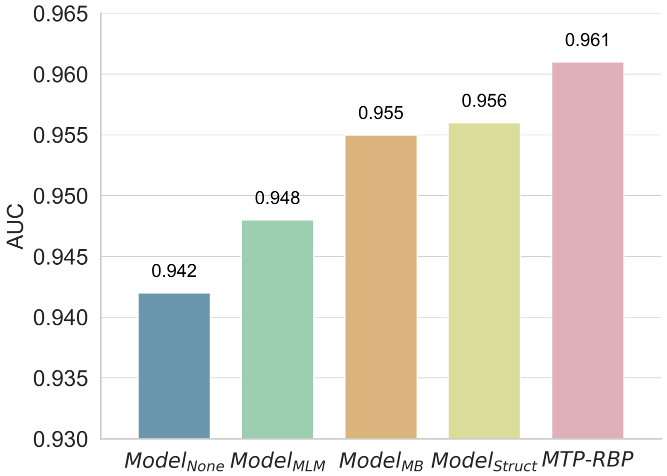

Fig. 4. AUC values of MTP-RBP under different pre-training strategies and with structural features as another branch inputs.

2) Representation Strategies

To validate the enhanced efficacy of our BPE-inclusive K-mer Semantic Representation strategy, we conducted a comparative analysis with k-mer embedding encoding ($\text{Model}_{\text{emb}}$) and k-mer semantic encoding ($\text{Model}_{\text{sem}}$). The key distinction between these methods and MTP-RBP lies in the fact that $\text{Model}_{\text{emb}}$ does not engage in pre-training tasks, while $\text{Model}_{\text{sem}}$ does not utilize BPE during RNA structural representation. Instead of utilizing the dimensionality of BPE, $\text{Model}_{\text{sem}}$ predicts dimensions for structural construction tasks at k^5.

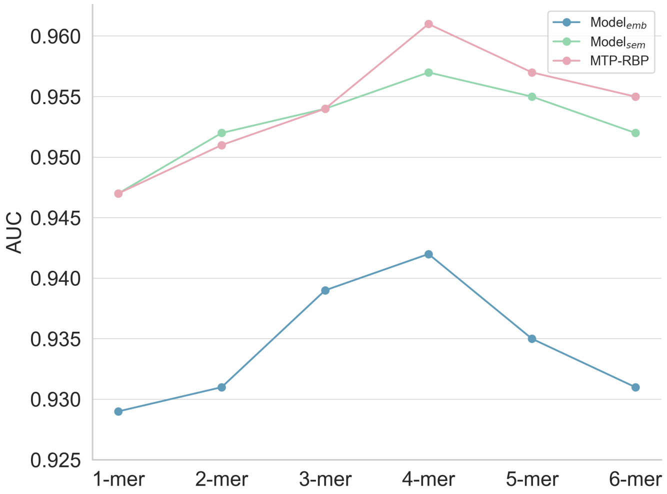

Fig. 5. The AUC of three representation strategies for various k-mer lengths.

From Fig.5, it can be seen that BPE-inclusive K-mer Semantic Representation strategy has the highest performance. In this case, regardless of the value of k, the AUC values of the k-mer embedding encoding-only method are lower than those of the method with pre-training. At k values of 4 and 6, MTP-RBP outperforms Model$_{\text{emb}}$ by 1.8% and 2.4% respectively, owing to MTP-RBP's execution of three pretraining tasks to learn deep representations of RNA sequences. This highlights the importance of enhancing encoding through pretraining. For Model$_{\text{sem}}$, it can be observed that when the value of k ranges from 1 to 3, it achieves performance comparable to MTP-RBP, and even slightly outperforms MTP-RBP when k equals 2. This phenomenon arises from the fact that when k is small, the prediction space dimensionality of RNA secondary structures is similarly reduced, rendering BPE unable to effectively capture these patterns and thus failing to fully leverage its advantages. When k exceeds 4, the AUC values of MTP-RBP consistently surpass those of Model$_{\text{sem}}$. This observation suggests that BPE is more adept at capturing the internal structure of the vocabulary and reducing the sparsity of the data when there is a large variety of subsequences, thereby favoring the model's performance in structural construction pre-training tasks. In summary, our proposed BPE-inclusive K-mer Semantic Representation strategy ensures the representation quality of RNA sequences by selecting appropriate k-values as well as effective encoding methods.

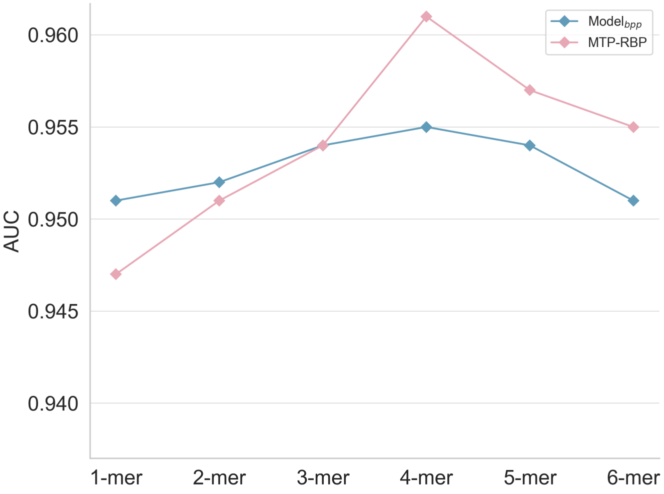

Fig. 6. AUC values of two structural representations across various k-mer lengths. Model$_{bpp}$ represents RNA secondary structures using base-pair probability matrices, while MTP-RBP employs motif annotations for secondary structure representation.

Furthermore, we explored the impact of two different RNA secondary structure representations—motif annotation and base-pair probability matrices—on model performance [37]. The performance of these two representations was compared across different k-mer lengths, as shown in Fig.6. When the k-mer length is less than 3, the model tends to perform better with base-pair probability matrices. This suggests that for shorter k-mers, the global pairing information provided by base-pair probability matrices can effectively compensate for the limitations of motif annotation in extracting structural features within short sequences. As the k-mer length increases, k-mers provide richer contextual information, allowing the model to better capture local structural patterns, which enhances motif annotation and improves downstream performance. When $k = 4$, the balance between local and global information reaches an optimal point, resulting in the highest overall model performance.

3) Model Architectures

To validate the efficacy of the model architecture, we conducted separate ablation experiments on the encoding stage architecture and the classification stage architecture.

a) Encoding Module

MTP-RBP modifies the base BERT architecture to suit RNA-protein interaction tasks. Specifically, MTP-RBP replaces the learnable position encoding with fixed position encoding to reduce the overall number of model parameters. Additionally, it omits the [CLS] token and instead employs all output vectors for classification in downstream tasks. To assess the efficacy of the model architecture modifications, we conducted an experiment with the following setup: We sequentially adjusted the MTP-RBP model architecture to include the [CLS] token (Model$_{\text{CLS}}$), employ learnable positional coding (Model$_{\text{POS}}$), and incorporate both the [CLS] token and learnable positional coding (Model$_{\text{CLSPOS}}$).

Fig.7 illustrates the AUC scores achieved by MTP-RBP models utilizing various architectures on RBP-24. The MTP-RBP model demonstrates the highest performance, followed by the model with learnable position encoding, whereas the remaining two models that introduce [CLS] token exhibit poor performance. RNA sequences are longer than natural language sequences, and it is difficult to characterize key features of longer sequences using only a single output vector corresponding to a [CLS] token. In summary, two enhancements of the MTP-RBP model over the basic BERT architecture prove effective.

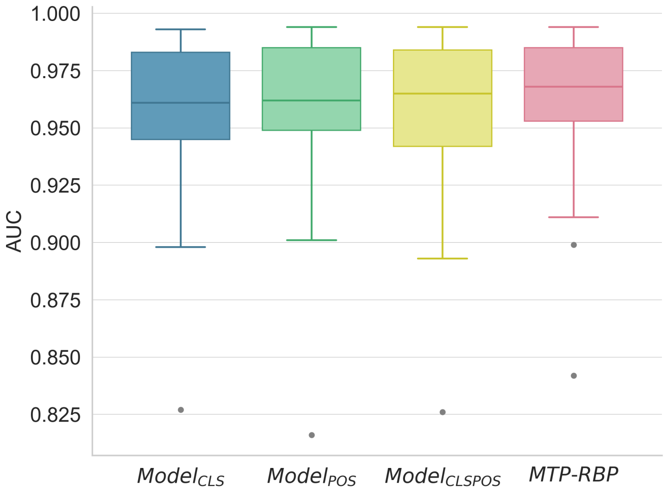

Fig. 7. The AUC values of the MTP-RBP models with different architectures on RBP-24.

b) Classification Module

The purpose of this section is to evaluate the plausibility of the binding function prediction module, which integrates a

BiGRU layer to the binary classifier to further capture long-range dependencies in sequences. As shown in Fig.8, Based on the MTP-RBP model, we construct two Variants by removing the BiGRU layer (Model$_1$), and replacing the BiGRU with a unidirectional GRU layer (Model$_2$), respectively. Additionally, we assess the performance of the frequently used BLSTM (Model$_3$) and CNN-BLSTM (Model$_4$).

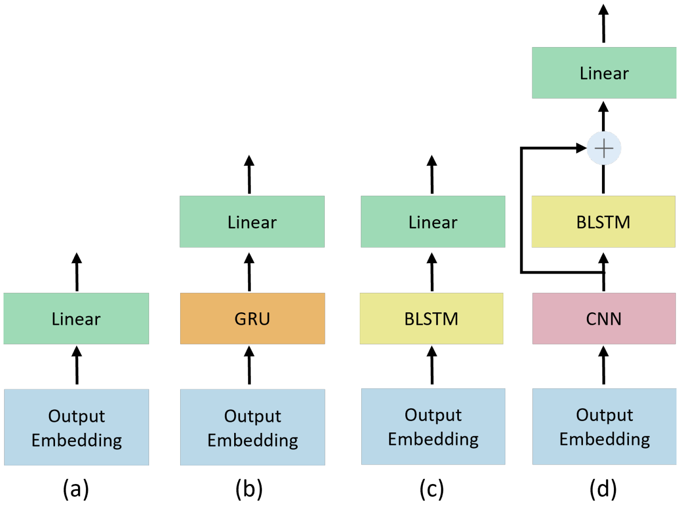

Fig. 8. Comparison models utilizing different modules in the classification phase.

TABLE VI
PERFORMANCE COMPARISON OF USING DIFFERENT MODULES IN THE CLASSIFICATION STAGE

Model	AUC	AP	Recall	MCC
Model$_1$	0.955	0.954	0.902	0.810
Model$_2$	0.957	0.957	0.904	0.813
Model$_3$	0.958	0.957	**0.911**	0.812
Model$_4$	0.957	0.956	0.902	0.810
MTP-RBP	**0.961**	**0.961**	0.908	**0.816**

***Bold**: best results.*

Table VI demonstrates that MTP-RBP outperforms the other four models in terms of AUC, AP, and MCC. Model$_1$ exhibits the lowest values across all four metrics when using only the binary classifier, compared to the models combined with GRU and LSTM. This suggests that GRU and LSTM are effective in further handling long-term dependencies, enabling the model to better memorize and utilize the encoded information. Additionally, comparing MTP-RBP and Model$_2$ shows that using BiGRU in the model achieves better performance than using GRU alone, as GRU only considers forward information in the RNA sequence, failing to capture the complex structure adequately. In contrast, BiGRU considers both forward and backward information, providing a more comprehensive sequence representation. Furthermore, although both MTP-RBP and Model$_3$ can capture long-term dependencies, GRU's gating mechanism may be more effective in balancing remembering and forgetting in some cases, particularly when dealing with long RNA sequences. Finally, comparing Model$_3$ and Model$_4$, we find that although the CNN-BLSTM architecture is effective in Protein-binding RNA prediction, using CNNs after high-quality encoding may cause a loss of information transfer, hindering the BLSTM's ability to capture essential features. In conclusion, our proposed feature prediction module for BiGRU-binary classifiers is effective.

IV. Conclusion

In this study, we propose a *pre-training - fine-tuning* model called MTP-RBP for predicting Protein-binding RNA. In order to learn the contextual deep representation of RNA sequences, we employed the Masked Language Modeling and devised a pre-training task integrating Binding Function Prediction to characterize the functional semantic information of k-mer sequences in various contexts. Furthermore, unlike current approaches that separately extract RNA sequence and structural features, our model incorporates a Structural Construction Model, enabling the Encoder to achieve better feature representation and fusion of sequence-structure-function features. The experimental results demonstrate that MTP-RBP achieves state-of-the-art performance on both the RBP-24 dataset and the ENCODE dataset, outperforming established models such as iDeepV, iDeepS, DeepCLIP, SA-Net, and WVDL. This outcome serves as compelling evidence of the efficacy of our proposed model.

Despite the promising performance of MTP-RBP, there remains considerable potential for improvement. As RNA structure prediction advances, integrating more accurate secondary and tertiary structure information could enhance the structure construction pre-training task. Additionally, in terms of functional characterization, our model focuses mainly on RNA binding features. Since RNA-protein interactions are influenced by various functional aspects, incorporating more diverse pre-training tasks could provide a fuller functional characterization. These improvements could further boost MTP-RBP's performance. Moreover, this sequence-structure-function integrated framework holds promise for broad application across various biological domains.

References

[1] M. Ramanathan, D. F. Porter, and P. A. Khavari, "Methods to study RNA–protein interactions," *Nat. Methods*, p. 225–234, Mar 2019.

[2] M. T. Weirauch *et al.*, "Evaluation of methods for modeling transcription factor sequence specificity," *Nat. Biotechnol.*, vol. 31, no. 2, p. 126–134, Feb 2013.

[3] J. Yan, S. Friedrich, and L. Kurgan, "A comprehensive comparative review of sequence-based predictors of DNA- and RNA-binding residues," *Brief. Bioinf.*, p. 88–105, Jan 2016.

[4] S. He, E. Valkov, S. Cheloufi, and J. Murn, "The nexus between RNA-binding proteins and their effectors," *Nat. Rev. Genet.*, vol. 24, no. 5, pp. 276–294, 2023.

[5] M. Corley, M. C. Burns, and G. W. Yeo, "How RNA-binding proteins interact with RNA: molecules and mechanisms," *Mol. Cell*, vol. 78, no. 1, pp. 9–29, 2020.

[6] E. Jankowsky and M. E. Harris, "Specificity and nonspecificity in RNA–protein interactions," *Nat. Rev. Mol. Cell Biol.*, vol. 16, no. 9, p. 533–544, Sep 2015.

[7] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning," *Nature Biotechnol.*, vol. 33, no. 8, pp. 831–838, 2015.

[8] D. Quang and X. Xie, "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences," *Nucleic Acids Res.*, vol. 44, no. 11, pp. e107–e107, 2016.

[9] A. G. B. Grønning *et al.*, "DeepCLIP: predicting the effect of mutations on protein–RNA binding with deep learning," *Nucleic Acids Res.*, vol. 48, no. 13, pp. 7099–7118, 2020.

[10] Z. o. Pan, "MCNN: Multiple Convolutional Neural Networks for RNA-Protein Binding Sites Prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 2, pp. 1180–1187, 2023.

[11] Z. Pan *et al.*, "WVDL: weighted voting deep learning model for predicting RNA-protein binding sites," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2023.

[12] X. Wang, M. Zhang, C. Long, L. Yao, and M. Zhu, "Self-Attention Based Neural Network for Predicting RNA-Protein Binding Sites," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 2, pp. 1469–1479, 2023.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome," *Bioinformatics*, vol. 37, no. 15, pp. 2112–2120, 2021.

[15] K. Yamada and M. Hamada, "Prediction of RNA–protein interactions using a nucleotide language model," *Bioinformatics Advances*, vol. 2, no. 1, p. vbac023, 2022.

[16] J. Chen *et al.*, "Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions," *arXiv preprint arXiv:2204.00300*, 2022.

[17] X. Pan, P. Rijnbeek, J. Yan, and H.-B. Shen, "Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks," *BMC genomics*, vol. 19, pp. 1–11, 2018.

[18] J. Wang *et al.*, "EDLMFC: an ensemble deep learning framework with multi-scale features combination for ncRNA–protein interaction prediction," *BMC Bioinf.*, vol. 22, pp. 1–19, 2021.

[19] Z. Shen, Q. Zhang, K. Han, and D.-s. Huang, "A Deep Learning Model for RNA-Protein Binding Preference Prediction based on Hierarchical LSTM and Attention Network," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, p. 1–1, Jan 2020.

[20] Y. Liu, W. Gong, Z. Yang, and C. Li, "SNB-PSSM: A spatial neighbor-based PSSM used for protein–RNA binding site prediction," *J. Mol. Recognit.*, vol. 34, no. 6, p. e2887, 2021.

[21] Y. Liu, W. Gong, Y. Zhao, X. Deng, S. Zhang, and C. Li, "aPRBind: protein–RNA interface prediction by combining sequence and I-TASSER model-based structural features learned with convolutional neural networks," *Bioinformatics*, vol. 37, no. 7, pp. 937–942, 2021.

[22] W. Wang *et al.*, "MAHyNet: Parallel Hybrid Network for RNA-Protein Binding Sites Prediction Based on Multi-Head Attention and Expectation Pooling," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, no. 01, pp. 1–12, 2024.

[23] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *Chinese Computational Linguistics*. Cham: Springer, 2019, pp. 194–206.

[24] D. Maticzka, S. J. Lange, F. Costa, and R. Backofen, "GraphProt: modeling binding preferences of RNA-binding proteins," *Genome Biol.*, p. R17, Jan 2014.

[25] Gerstberger, Stefanie and Hafner, Markus and Tuschl, Thomas, "A census of human RNA-binding proteins," *Nat. Rev. Genet.*, vol. 15, no. 12, pp. 829–845, 2014.

[26] G. Anders *et al.*, "doRiNA: a database of RNA interactions in post-transcriptional regulation," *Nucleic Acids Res.*, vol. 40, no. D1, p. D180–D186, Jan 2012.

[27] X. Pan, Y. Fang, X. Li, Y. Yang, and H.-B. Shen, "RBPsuite: RNA-protein binding sites prediction suite based on deep learning," *BMC genomics*, vol. 21, pp. 1–8, 2020.

[28] E. L. Van Nostrand *et al.*, "A large-scale binding and functional map of human RNA-binding proteins," *Nature*, vol. 583, no. 7818, pp. 711–719, 2020.

[29] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *arXiv preprint arXiv:1508.07909*, 2015.

[30] A. R. Gruber, R. Lorenz, S. H. Bernhart, R. Neuböck, and I. L. Hofacker, "The vienna RNA websuite," *Nucleic Acids Res.*, vol. 36, no. suppl_2, pp. W70–W74, 2008.

[31] Y. Zhu, Z. Xie, Y. Li, M. Zhu, and Y.-P. P. Chen, "Research on folding diversity in statistical learning methods for RNA secondary structure prediction," *Int J Biol Sci.*, vol. 14, no. 8, p. 872, 2018.

[32] C. B. Do, D. A. Woods, and S. Batzoglou, "CONTRAfold: RNA secondary structure prediction without physics-based models," *Bioinformatics*, vol. 22, no. 14, pp. e90–e98, 2006.

[33] L. Huang, H. Zhang, D. Deng, K. Zhao, K. Liu, D. A. Hendrix, and D. H. Mathews, "LinearFold: linear-time approximate RNA folding by 5'-to-3'dynamic programming and beam search," *Bioinformatics*, vol. 35, no. 14, pp. i295–i304, 2019.

[34] T. Binet, S. Padiolleau-Lefèvre, S. Octave, B. Avalle, and I. Maffucci, "Comparative study of single-stranded oligonucleotides secondary structure prediction tools," *BMC Bioinf.*, vol. 24, no. 1, p. 422, 2023.

[35] A. Vaswani *et al.*, "Attention is all you need," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 30, 2017.

[36] X. Pan and H.-B. Shen, "Learning distributed representations of RNA sequences and its application for predicting RNA-protein binding sites with a convolutional neural network," *Neurocomputing*, vol. 305, pp. 51–58, 2018.

[37] D. H. Mathews, "Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization," *Rna*, vol. 10, no. 8, pp. 1178–1190, 2004.

Lin Gan received the BS degree in computer science and technology from the College of Computer Science, Sichuan University, in 2023. He is currently working toward an MS degree in the College of Computer Science, at Sichuan University. His research interests include bioinformatics and deep learning.

Xinyi Wang received his BS degree in computer science and technology from the College of Computer Science, Chongqing University, in 2020, and his MS degree in Computer Science and Technology from Sichuan University, in 2023. His research interests include bioinformatics and deep learning.

Yi Zhou received her BS degree in computer science and technology from the College of Computer Science, Sichuan University, in 2021, and her MS degree in Computer Science and Technology from Sichuan University, in 2024. Her research interests include AI4Science, large language models, and graph machine learning.

Min Zhu received the PhD degree in applied mathematics from Sichuan University, in 2004. She is currently a professor at the College of Computer Science, Sichuan University, and has presided over a number of national and provincial research projects. Based on the research works, she has published more than 100 academic papers in journals and conferences. Her current research interests include bioinformatics, visual analysis, and image processing.