Mathematical Biosciences and Engineering

MBE, 21(1): 272–299.
DOI: 10.3934/mbe.2024013
Received: 24 August 2023
Revised: 22 November 2023
Accepted: 28 November 2023
Published: 12 December 2023

http://www.aimspress.com/journal/MBE

Research article

MTTLm^{6}A: A multi-task transfer learning approach for base-resolution mRNA m^{6}A site prediction based on an improved transformer

Honglei Wang[1,2], Wenliang Zeng[1], Xiaoling Huang[1], Zhaoyang Liu[1], Yanjing Sun[1,*] and Lin Zhang[1,*]

[1] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China
[2] School of Information Engineering, Xuzhou College of Industrial Technology, Xuzhou, China

* **Correspondence:** Email: yjsun@cumt.edu.cn, lin.zhang@cumt.edu.cn; Tel: +86 0516-83592156, +8613605213263.

Abstract: N6-methyladenosine (m^{6}A) is a crucial RNA modification involved in various biological activities. Computational methods have been developed for the detection of m^{6}A sites in Saccharomyces cerevisiae at base-resolution due to their cost-effectiveness and efficiency. However, the generalization of these methods has been hindered by limited base-resolution datasets. Additionally, RMBase contains a vast number of low-resolution m^{6}A sites for Saccharomyces cerevisiae, and base-resolution sites are often inferred from these low-resolution results through post-calibration. We propose MTTLm^{6}A, a multi-task transfer learning approach for base-resolution mRNA m^{6}A site prediction based on an improved transformer. First, the RNA sequences are encoded by using one-hot encoding. Then, we construct a multi-task model that combines a convolutional neural network with a multi-head-attention deep framework. This model not only detects low-resolution m^{6}A sites, it also assigns reasonable probabilities to the predicted sites. Finally, we employ transfer learning to predict base-resolution m^{6}A sites based on the low-resolution m^{6}A sites. Experimental results on Saccharomyces cerevisiae m^{6}A and Homo sapiens m^{1}A data demonstrate that MTTLm^{6}A respectively achieved area under the receiver operating characteristic (AUROC) values of 77.13% and 92.9%, outperforming the state-of-the-art models. At the same time, it shows that the model has strong generalization ability. To enhance user convenience, we have made a user-friendly web server for MTTLm^{6}A publicly available at http://47.242.23.141/MTTLm6A/index.php.

Keywords: RNA modification site; multi-task learning; transfer learning; natural language processing; deep learning

1. Introduction

It has been demonstrated that more than 170 types of RNA modifications are reported within a diverse set of RNAs [1], including m^6A, adenosine to inosine (A-to-I) deamination, cytosine to uracil (C-to-U) deamination, N^1-methyladenosine (m^1A), 5-methylcytosine (m^5C), pseudouridylation (Ψ), and ribose 2'O-methylation [2], etc. There is a growing list of RNA modifications found in both coding and non-coding RNAs, significantly influencing their biological functions [3]. These modifications frequently result in changes to RNA stability, folding, interactions, translation, localization, and subsequent processing, thereby impacting their biological function [4,5]. However, insights into the molecular machineries responsible for the deposition and removal, as well as recognition and interpretation, of these modifications within the cell are available for only a few modifications [6]. Even for those modifications for which writers, erasers, and readers have been identified [7], such as m^6A [8], we have limited knowledge about their regulation, their cooperation or competition with other RNA modification and processing events, and how they become deregulated in disease [9]. Therefore, the accurate identification of m^6A sites is a crucial step in understanding the mechanisms underlying these biological phenomena [9].

To date, several experimental methods have been developed to localize the m^6A site. High-throughput sequencing technologies have been successfully applied to detect m^6A sites in various species, including Saccharomyces cerevisiae [10], Homo sapiens [11,12], Arabidopsis thaliana [13], and mouse [14]. However, it is important to note that most high-throughput sequencing techniques cannot quickly obtain and precisely pinpoint the exact location of the m^6A site [15]. The m^6A motif 'RRACH' is often used to further narrow the location of the m^6A site to a basic resolution within the peak detected with the m^6A signal. Other experimental techniques, such as miCLIP-seq [16], can identify m^6A sites at the single nucleotide resolution level. However, these methods rely on m^6A-specific antibodies, exhibit poor reproducibility are long and involve complex procedures [17], making them unsuitable for large-scale genomic data analysis. Hence, there is a strong motivation to explore computational methods that can accurately and efficiently identify methylation sites.

Researchers have developed a variety of computational methods to predict RNA modification sites, which serve as invaluable complements to experimental approaches [18]. These methods approach RNA methylation identification as a binary prediction task [19], training machine learning models to differentiate between truly methylated and unmethylated sites. By leveraging these computational methods, we can quickly predict whether this sequence contains RNA methylation sites. The traditional computational method involves extracting a comprehensive set of hand-designed features from biological sequences [20]. These features are then fed into classical shallow classification algorithms [21], such as a support vector machine [22], often utilizing a linear kernel. However, the selection of these features is typically an empirical process, relying on trial and error [23]. Moreover, the feature selection itself is highly task-dependent, necessitating additional research for each new predictive task [24].

It is evident that analyzing biological sequences and interpreting the underlying biological information pose significant challenges in the realization of biological breakthrough discoveries. Recently, the application of natural language processing (NLP) in sequence analysis has garnered considerable attention within the realm of biological sequence processing [25]. This approach

considers biological sequences as sentences [21,26], while k-mer subsequences are akin to words [25,27]; NLP has emerged as a valuable approach for unraveling the structure and function encoded within these sequences [28,29]. In contrast to traditional machine learning methods, deep learning (DL) techniques offer an end-to-end design [30]. The input sentence undergoes a series of feature extraction layers, with the deep layers of the network automatically learning features that are relevant to the task through backpropagation [31]. The journey from raw data to the ultimate output entails the extraction of features derived directly from the input data, honing in on the crucial aspects for the final identification or prediction tasks. This intricate process involves sifting through the raw information, meticulously selecting the most pertinent characteristics, and transforming them into more meaningful representations. These extracted features serve as the bedrock of the entire analytical process, empowering the algorithms and models to discern patterns and relationships, which is essential for achieving accurate and insightful conclusions. Through this transformative journey, the system gains the ability to decipher complex information and deliver informed decisions, ultimately driving successful outcomes [32]. For example, EDLm^{6}Apred [32] applies bidirectional long short-term memory (BiLSTM) to predict m^{6}A site through the use of word2vec [33], RNA word embedding [34] and one-hot [35] encoding. However, long short-term, BiLSTM and recurrent neural networks do not allow parallel computation [36], which results in long training times [37]. Convolutional neural network (CNN) has the ability to achieve parallel computation [38] and learn local dependencies [39]. For example, in the case of m^{6}A-word2vec [40], a CNN is employed to identify m^{6}A sites by extracting features based on word2vec. Similarly, Deeppromise [41] utilizes a CNN to identify m^{1}A and m^{6}A sites, extracting features through integrated enhanced nucleic acid composition [42,43], one-hot encoding, and RNA word embedding. However, these CNN structures primarily focus on contextual relationships among neighboring bases [44], without taking into account the dependencies over long distances within the sequence [45]. To address this limitation, DeepM^{6}ASeq [46] combines the strengths of CNNs and BiLSTM by incorporating two layers of a CNN and one layer of BiLSTM to predict m^{6}A sites. While this approach can be effective, it may extract redundant features that can interfere with prediction performance [47]. To quantify the degree of word-to-word dependency, the attention mechanism comes into play. By applying the attention mechanism, it becomes possible to capture the specific words that significantly impact the classification results. MultiRM [48], on the other hand, employs a BiLSTM layer and a Bahdanau attention [49] layer to identify various types of RNA modification sites and extract features based on word2vec encoding. In this case, Bahdanau attention calculates the attention weights for two words in different sentences. However, since Google introduced the transformer model in 2018, self-attention has been recognized as a special case of the attention mechanism [50,51]. The Transformer model, based entirely on self-attention mechanisms, has become the most widely used architecture in NLP representation learning, as demonstrated by its adoption in various applications [52]. Among them, Plant^{6}mA [53], a transformer encoder, can be employed to identify whether the input sequence contains an m^{6}A site. However, in the context of the transformer model, positional encoding plays a vital role, as other key components of the model are completely invariant to the order of the sequence. The original transformer employs absolute positional encoding, assigning each position a unique embedding vector [50]. By adding the positional embedding to the word embedding, the model affords valuable insights into the contextual representations of words at different positions. In addition to absolute positional encoding, Shaw et al. [54] and Raffel et al. [55]. have introduced relative positional encoding. This innovative technique incorporates carefully designed bias terms within the self-attention module, enabling the encoding of

the distance between any two positions. However, it has been demonstrated by Ke et al. [56] that the additional operation applied to positional encoding and word embeddings in absolute positional encoding can introduce mixed correlations and unnecessary randomness in the attention mechanism. This may limit the expressive power of the model, potentially impacting its performance. Furthermore, the feed-forward networks within the transformer structure struggle to effectively capture contextual information. Since position-wise feed-forward networks process each position independently, they lack the capacity to adequately capture global contextual information. Consequently, the model may face difficulties in accurately comprehending long-term dependencies or recognizing global patterns within the sequence.

As m^6A is the most prevalent modification observed in mammals, numerous methods have been developed to predict m^6A sites in Saccharomyces cerevisiae. However, these methods [57–61] have primarily relied on a small dataset consisting of only 1307 m^6A sites, as derived from base-resolution sequencing. Unfortunately, the limited size of this dataset has hindered the full utilization of the advantages offered by DL methods [62]. However, RMBase [63,64] and m^6A-Atlas [65] respectively document over 60,000 low-resolution and 10,000 high-resolution m^6A sites in Saccharomyces cerevisiae. Regarding the relatively novel m^1A site prediction, it also encounters the above problems. Many methods for predicting m^1A are primarily based on a smaller dataset containing only 707 human m^1A sites with base-resolution sequencing. Correspondingly, RMBase has records of more than 2000 low-resolution human m^1A sites. Huang et al. [66] proposed WeakRM, the first weakly supervised learning framework for predicting RNA modifications from low-resolution epitranscriptome datasets, such as those generated from acRIP-seq and hMeRIP-seq. Astonishingly, these extensive datasets have not been fully leveraged for the development of computational methods in this context. In most scenarios, our primary concern revolves around achieving optimal performance on one task, which requires the training of a single model or an ensemble of models to perform our desired task, as well as the fine-tuning and optimization of models. Through this process, we continuously iterate and refine these models until they reach a point where their performance plateaus. While this approach often yields acceptable results, by focusing on a single task, it tends to overlook valuable information that could potentially enhance our desired metrics. Specifically, that information comes from the training signals derived from related tasks. By leveraging shared representations among these related tasks, we can empower our model to generalize better and improve its performance on the original task.

The corresponding the number of supporting experiments or studies (NSES) [63] information for the methylation sites in RMBase may be the key information mentioned above that is ignored. Intuitively, the larger the number of experimental identifications of a methylation site, the greater our confidence in considering it as a genuine methylation site. Currently, the exploration of multi-task prediction for methylation sites incorporating the NSES information is still in its early stages. An example of such an algorithm is MTDeepM6A-2S [67], which entails the use of the NSES information in the construction of a multi-task model based on a combination of a CNN and BiLSTM deep framework. This model was designed for the prediction of base-resolution m^6A sites. However, one limitation of the BiLSTM component lies in its sequential nature, where computations are executed step-by-step. As a result, the computational speed tends to be slower, and it becomes challenging to capture distant dependencies and global contextual information within the sequence. These factors can potentially limit the model's ability to effectively understand long-range relationships and extract comprehensive contextual features. Therefore, it is essential to assign relatively large attention weights to the vital information. While MTDeepM6A-2S represents a significant advancement in incorporating

NSES information into the multi-task prediction of methylation sites, there is still room for further improvement. Addressing the limitations associated with the sequential computations of BiLSTM and enhancing the capture of remote dependencies and global contextual information remain important areas for future research in this field.

To address the limitations of existing models, we drew inspiration from the multi-stage post-calibration determinations used for high-resolution m^6A site identification and the concept of multi-task learning. As a result, we propose A multi-task transfer learning approach for base-resolution mRNA m^6A site prediction based on an improved transformer (MTTLm^6A). In the initial stage, known as the source domain-stage, by using the NSES information for multi-task learning, we have improved the performance of the model in terms of the ability to detect low-resolution m^6A sites by optimizing the transformer model structure, specifically, the structure applies the double multi-head-attention (multi-head-attention+multi-head-attention) mechanism, which assigns relatively large attention weights to the critical information to intensify it. In the target domain-stage, considering the similarity between the classification tasks in both stages, we have transferred the weights of specific layers and deep networks from the model trained in the source domain-stage to the model in the target domain-stage to predict m^6A sites at base-resolution. Experimental results on Saccharomyces cerevisiae m^6A and Homo sapiens m^1A data demonstrate that MTTLm^6A achieves area under the receiver operating characteristic (AUROC) values of 77.13% and 92.9%, outperforming the state-of-the-art models. At the same time, it shows that the model has strong generalization ability. To enhance user convenience, we have made a user-friendly web server for MTTLm^6A publicly available at http://47.242.23.141/MTTLm6A/index.php.

2. Materials and methods

2.1. Benchmark datasets

We extracted datasets of two major types of RNA modification sites, including m^1A and m^6A, from the RMBase v2.0. For the m^1A sites, we collected low-resolution m^1A sites of Homo sapiens from the extensive database RMBase v2.0, in which 2574 m^1A sites have been recorded. The RNA segments with upstream and downstream nucleotides were obtained from the genome. Negative sites (non-modified nucleotides) were randomly selected from the unmodified bases of the same transcript containing the positive sites. The negative samples were down-sampled and cut short to match the number and size of the positive samples. To avoid overfitting, CD-HIT [68] was used with a threshold of 0.7 to remove redundant segments. The redundancies of positive and negative samples were removed. Thus, we got 1987 positive samples and 2249 negative samples. To obtain a balanced dataset, 1987 negative samples were randomly selected to build the final dataset. For the second-stage model, we collected base-resolution m^1A sites of Homo sapiens from DeepPromise [69] as positive samples. Consequently, 593 training samples and 114 test samples were obtained. Because the second-stage model is used to identify base-resolution m^1A sites from low-resolution m1A sites, we used the low-resolution m^1A sites recorded in RMBase 2.0 as negative samples in the current study. Therefore, based on the above 1987 positive samples, 707 (593 + 114) positive samples were randomly assigned to the second-stage as negative samples; and the remaining samples were divided into training sets and independent test sets at a ratio of 4:1 for the first stage model.

For the m^6A sites, the dataset was derived from the low-resolution m^6A sites previously described

by Wang et al. [67]. This dataset contains a total of 24,669 m^6A sites. Within these segments, two distinct central motif patterns exist, i.e., AAC and GAC. Notably, the existing methods for predicting m^6A sites in Saccharomyces cerevisiae were developed by using the Met2614 dataset [57], which only includes the GAC central motif. To ensure a comprehensive analysis, we divided the original RNA segments into two parts: one containing segments with the GAC central motif and the other containing segments with the AAC central motif. The number of segments with the AAC central motif was 13,732, while the number of segments with the GAC central motif was 10,937. The ratio of positive to negative samples in both datasets was 1:1. Subsequently, the datasets were randomly split into benchmark and independent test datasets at a 4:1 ratio. This resulted in the AAC benchmark dataset containing 10,985 positive and negative samples, and the independent test dataset containing 2747 positive and negative samples. Similarly, the GAC benchmark dataset consisted of 8749 positive and negative samples, and the independent test dataset contained 2188 positive and negative samples. Referring to the experimental results in Chen's [69] and Wang's [67] paper, the sizes of the optimal window were respectively set as 101nt and 601nt for m^1A and m^6A sites.

Furthermore, in RMBase v2.0, the NSES value associated with each m^6A site recorded in the database was used as the target for a regression task. The NSES indicates the number of experimental confirmations for the corresponding adenine being modified [63]. In simpler terms, a higher NSES value suggests a higher level of certainty regarding the authenticity of the m^6A modification at that specific site. The distribution of NSES within the m^6A dataset is depicted in Figure 1.

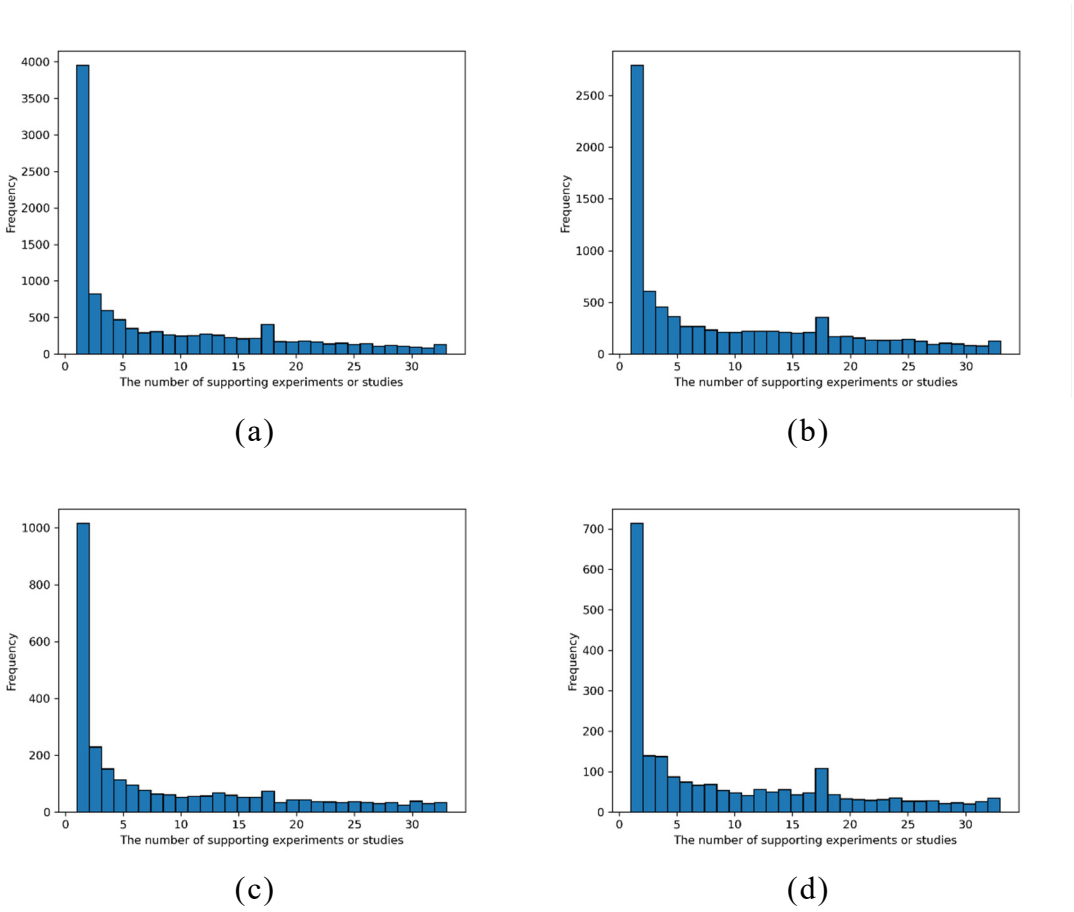

Figure 1. Histograms of NSES on the m^6A datasets. (a) AAC_BM, (b) GAC_BM, (c) AAC_IND, (d) GAC_IND.

For the target domain-stage model, the positive samples were obtained from the base-resolution m^6A sites of Saccharomyces cerevisiae from m^6A-Atlas [65]; 4689 m^6A sites were obtained, and they are all with GAC in the central motif. On the other hand, the negative samples were selected from the low-resolution m^6A sites with GAC in the central motif, as recorded in RMBase v2.0. The ratio of positive to negative samples in both datasets was set at 1:1 to ensure a balanced training environment. Similar to the source domain-stage model, the datasets were randomly divided into benchmark datasets and independent test datasets at a 4:1 ratio. The statistics of the datasets are shown in Table 1.

Table 1. Statistics of the benchmark and independent test datasets.

Site	Species	Stage	Datasets	Window size	Number of positive	Number of negative
m^6A	Saccharomyces cerevisiae	the source domain-stage	AAC_BM	601	10,985	10,985
			AAC_IND	601	2747	2747
			GAC_BM	601	8749	8749
			GAC_IND	601	2188	2188
		the target domain-stage	GAC_hr_BM	601	3751	3751
			GAC_hr_IND	601	938	938
m^1A	Homo sapiens	the source domain-stage	BM	101	1024	1024
			IND	101	256	256
		the target domain-stage	hr_BM	101	593	593
			hr_IND	101	114	114

BM benchmark; IND independent

2.2. Encoding of RNA segments

In the development of highly accurate computational methods, the features of sequence data play a crucial role. Suppose that we have raw data $R_0 = \{x^m\}_{m=1}^{M}$, where M is the number of sequences and each $x^m \in \mathbb{R}^{l_0}$ is an RNA sequence. Each entry x_i^m, $i = 1, 2, 3, 4, ..., l_0$ at position i takes its value from the alphabet $\sum = \{A, U, C, G, N\}$ from a sequence of constant length l_0.

One widely used and effective encoding method is one-hot encoding, which provides a simple yet powerful approach to representing the RNA sequences. In this method, the four nucleotides (*A, U, C, G*) are encoded as binary vectors: A = (1, 0, 0, 0), U = (0, 1, 0, 0), C = (0, 0, 1, 0), G = (0, 0, 0, 1), and N = (0, 0, 0, 0), representing unknown or ambiguous positions. After that, $R_0 = \{x^m\}_{m=1}^{M}$, where each $x^m \in \mathbb{R}^{l_0 \times 4}$ is an RNA sequence. By applying this encoding scheme, a sequence of 601 nucleotides is transformed into a matrix of 601 × 4.

2.3. Deep network and transfer learning

We have devised a multi-task transfer learning approach for base-resolution mRNA m^6A site prediction based on an improved transformer. The structure of the model is shown in Figure 2, and it is divided into the target source stage and the target domain-stage. The details are as follows.

The conventional approaches employed for the prediction of RNA m^6A sites have primarily relied

on single-task learning methods for classification. In contrast, we have adopted a novel multi-task architecture in the construction of the source domain model. Our aim is to enhance the classification results and realize a reasonable confidence value. To achieve this, we constructed a regression task by using the invaluable NSES information retrieved from RMBase v2.0. This regression task allows us to assign a confidence score to the classification results, thereby enhancing their interpretability and overall reliability. The feature encoding sequences were fed into the CNN layer in order to capture sequence patterns or motifs; the mathematical formulation of the CNN model is given below:

$$Conv(R)_{jf} = ReLU(\sum_{d=0}^{D-1}\sum_{n=0}^{N-1} W_{dn}^{f} R_{j+d,n}) \tag{1}$$

where R denotes the input matrix, f represents the index of the kernel, and j represents the index of the output position; each filter W^f is a $D \times N$ weight matrix, where D is the filter size, and N is the input channels; $\mathbb{R}^{l_0 \times 4} \mapsto \mathbb{C}^{l \times d}, l = l_0 - f + 1$.

Since convolution contains the order of the sequence, to avoid the occurrence of mixed correlations that may arise if the model is connected to a positional encoding layer in the transformer, we intentionally omit the positional encoding layer and directly link it to the multi-head-attention mechanism, the calculation of attention in this module can be divided into three steps.

In the first step, linearly transform the output matrix X of the CNN layer and divide it into three matrices, as follows

$$Q = XW^{q}, K = XW^{k}, V = XW^{v} \tag{2}$$

where $X \in \mathbb{C}^{l \times d}, l = l_0 - f + 1$; then, three learnable matrices, W^q, W^k and W^v are used to project X into different spaces. Usually, the matrix size of each of the three matrices is $\mathbb{C}^{d \times d_k}$, where d_k is a hyper-parameter.

In the second step, the scaled dot product attention can be calculated by using the following equations:

$$A_{m,n} = Q_m K_n^T \tag{3}$$

$$Attention(Q, K, V) = softmax(\frac{A}{\sqrt{d_k}})V \tag{4}$$

where Q_m is the query vector for the m-th token and K_n is the key vector representation of the nth token. The Softmax function is applied along the last dimension. Instead of using one group of W^q, W^k, W^v, using several groups will enhance the ability of self-attention.

In the third step, when several groups are used, it is called multi-head self-attention; the calculation can be formulated as follows:

$$Q^{(h)} = XW^{q(h)}, K^{(h)} = XW^{k(h)}, V^{(h)} = XW^{v(h)} \tag{5}$$

$$head^{(h)} = Attention(Q^{(h)}, K^{(h)}, V^{(h)}) \tag{6}$$

$$MultiHead(Q, K, V) = Concat(head^{(1)}, head^{(2)}, ..., head^{(i)})W^{o} \tag{7}$$

where i is the number of heads and the superscript h represents the head index. Usually $d_k \times n = d$,

which means that the output of $\left[head^{(1)}, head^{(2)}, ..., head^{(i)}\right]$ will be of size $\mathbb{C}^{l\times d}$. Also note that $W^o \in \mathbb{C}^{d\times d}$, which is a learnable parameter.

Furthermore, to more effectively capture contextual information, we deliberately replaced the feed-forward layer in the transformer structure with the multi-head-attention mechanism layer. This choice empowers the model to assign greater attention weight to key information, thereby reinforcing its significance. At the heart of the source domain-stage lies the primary objective of classifying low-resolution m^{6}A sites and accurately distinguishing them from non-m^{6}A sites. This pivotal step forms the foundation for analyses and investigations in the subsequent target domain-stage.

Building upon the multi-task model training obtained in the source domain-stage, we progress to the target domain-stage. In this stage, we employ a transfer learning strategy to train the target domain-stage model and focus on identifying both base-resolution m^{6}A sites and low-resolution m^{6}A sites.

2.3.1. Deep learning for building the source domain-stage model

During the source domain-stage, our model takes RNA segment sequences and the NSES information as input; the RNA segment sequences are then transformed into numerical matrices by the one-hot encoding process, as shown in Figure 2. These numerical matrices are then fed into the deep network, which consists of a CNN and double-multi-head-attention, referred to as CNN+MM.

The CNN uses a 1D convolutional layer to extract local features from the input matrices. To optimize the hyperparameters, we employed a grid-search strategy. In this stage, we used 16 convolutional kernels, each with a size of 10. Subsequently, the output of the CNN stage is normalized with a group normalization layer, where the number of groups was set to 4. The multi-head-attention stage consists of two multi-head-attention networks. One attention mechanism has two heads, with each head having a size of 8 (d_model = 8). The other attention mechanism has four heads, with each head having a size of 16 (d_model = 16). To promote effective information flow, we incorporated a dropout layer and a residual connection around each of the two sub-layers. The dropout rate was set at 0.1 to reduce overfitting, and layer normalization was applied subsequently. Following the multi-head-attention stage, an AveragePooling1D layer is applied to reduce the dimensionality of the extracted features. The kernel size of the 1D pooling layer was set to 15. Subsequently, the data were flattened into a 1D form by using a flattening layer. This is followed by a dropout layer and a fully connected layer. The dropout rate was set at 0.6, and the fully connected layer was set to comprise 64 neurons activated by the exponential linear unit (ELU) function.

The output layer of our model consisted of two outputs, catering to the classification and regression tasks, respectively. The estimated loss is made up of classification loss and regression loss. The following calculation is then used to get the total loss:

$$loss_{multitask} = loss_{classification} + \lambda loss_{regression} \tag{8}$$

where $loss_{classification}$ is the classification loss, $loss_{regression}$ is the regression loss, and λ can be set arbitrarily according to specific circumstances. For the classification task, the Softmax activation function was employed, and the categorical cross-entropy was specified as the loss function. For the regression task, the activation function ELU was used, with the log-cosh employed as the loss function. Therefore, the overall loss function for the entire multi-task model can be expressed as follows:

$$loss_{multitask} = -\frac{1}{N}\sum_{i=1}^{N}(y_i^{class}\log p_i^{class} + (1-y_i^{class})\log(1-p_i^{class})) + \lambda\sum_{i=1}^{N}\log(\cosh(y_i^{regression} - p_i^{regression})) \tag{9}$$

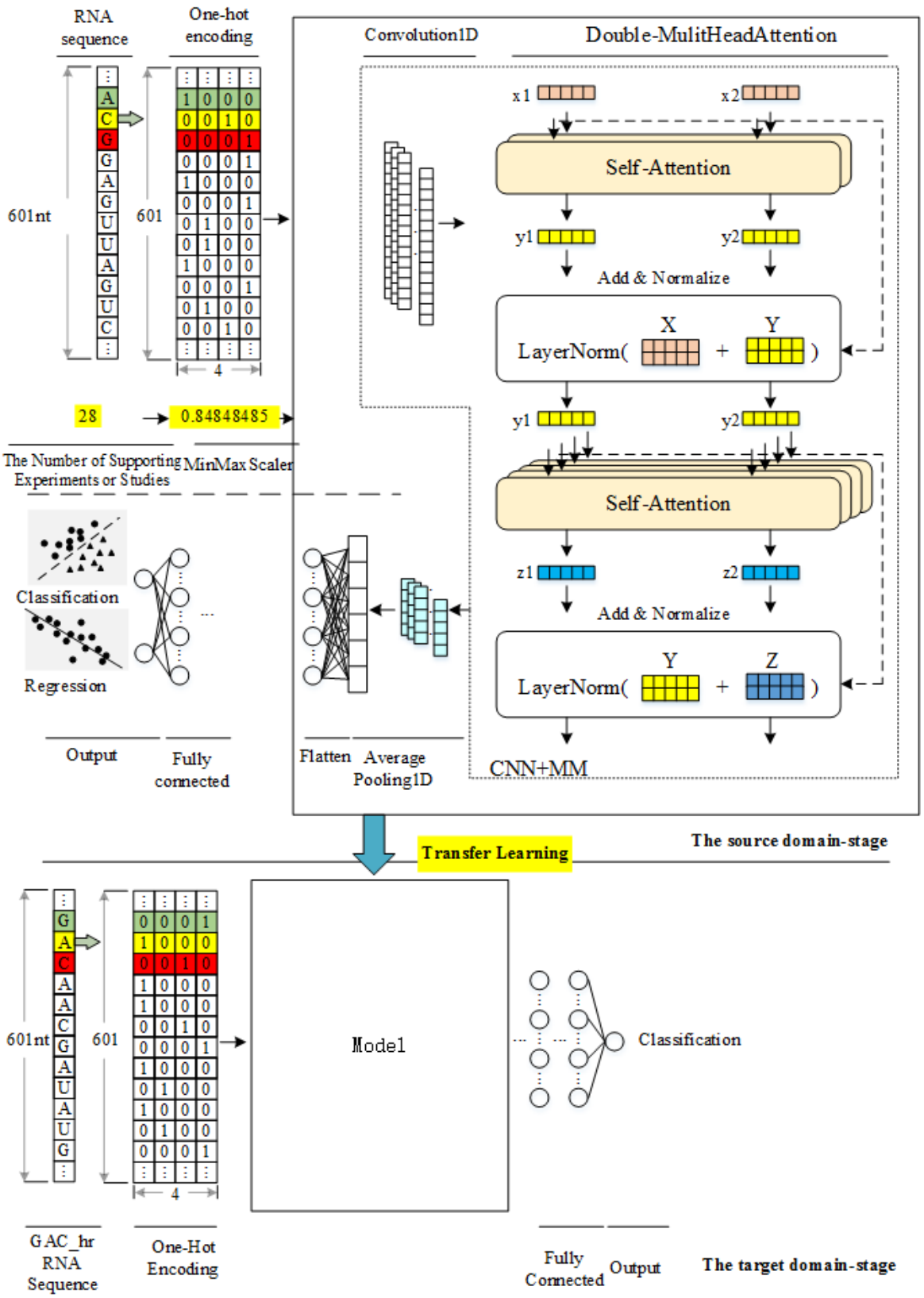

Figure 2. The diagram of the model. The source domain-stage model is used to discriminate low-resolution m^{6}A sites from non-m^{6}A sites, and the target domain-stage model is used to identify high-resolution m^{6}A sites from low-resolution m^{6}A sites.

where N denotes the total number of samples in the dataset and y_i^{class} and p_i^{class} denote the true label and prediction probability of the ith sample of the classification task, respectively. Similarly, $y_i^{regression}$ and $p_i^{regression}$ denoted the true label and prediction probability of the ith sample of the regression task, respectively. The total loss function is optimized by using a grid search with the weight parameter λ set to 0.6. This loss function leverages the label information from the regression task to potentially enhance the prediction accuracy of the classification task.

Finally, the stochastic gradient descent (SGD) optimization algorithm is used with the momentum set to 0.95 and a learning rate of 0.01. SGD is a widely adopted optimization algorithm known for its effectiveness in iteratively adjusting the model's parameters during the training process to minimize the loss function.

2.3.2. Transfer learning for the construction of the target domain-stage model

To construct the target domain-stage model, we used transfer learning by transferring the feature extraction layers from the source domain-stage model. This approach was motivated by the similarity between the classification tasks in both stages.

During the transfer learning process, we initialized the parameters of the target domain-stage model by using the feature extraction layers from the source domain-stage model. This initialization includes all layers except the output layer, ensuring that the model starts with valuable learned representations. By inheriting the corresponding weights, we have provided a strong foundation for the target domain model.

Subsequently, we optimized all of the weights of the target domain-stage model during training without freezing any layers. This allowed the model to adapt and refine its parameters based on the target domain's specific characteristics and data. By performing transfer learning in this way, we aimed to capitalize on the knowledge and patterns learned in the source domain, ultimately enhancing the performance and generalization of the model on the target domain's classification tasks.

2.3.3. Evaluation indicators

In this study, we comprehensively evaluated the performance of our prediction model by using eight commonly used classification metrics. These metrics include the accuracy (Acc), sensitivity (Sen), precision (Pre), Matthews correlation coefficient (MCC), specificity (Sp), and F1 score (F1). The formulas for these metrics are respectively as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

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

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

Additionally, we used the AUROC curve and the area under the precision-recall curve (AUPRC) to visually assess the overall performance of our model. These metrics provide insights into the model's ability to discriminate between different classes and the precision-recall trade-off, respectively. Time (s) indicates the training time of the model per epoch. By considering both the quantitative metrics and the visual evaluation, we gain a comprehensive understanding of the predictive capabilities of our model.

To evaluate the regression task, we used the Pearson correlation coefficient (*PCC*) as the index. The PCC measures the similarity between the predicted target values (X) and the actual target values (Y) of the samples. It is calculated as follows:

$$PCC = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{16}$$

Here, $\mathrm{cov}(X, Y)$ represents the covariance between X and Y, and σ_X and σ_Y represent the standard deviations of X and Y, respectively. The PCC ranges from –1 to 1, where a value of 0 indicates no correlation.

3. Results

3.1. Comparison with other different learning models based on benchmark dataset

By applying different values of the loss weight λ of the regression task (i.e., λ = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]), various models, including CNN, CNN+BiLSTM (CNN+BiL), CNN+transformer (CNN+TF), CNN+multi-head-attention+feed-forward (CNN+MF), and CNN+multi-head-attention+multi-head-attention (CNN+MM) models, were evaluated by using 5-fold cross-validation based on a benchmark dataset. Among them, the CNN+MF model directly connects the multi-head-attention mechanism layer and the feed-forward layer without connecting the positional encoding layer. On the other hand, the CNN+MM model replaces the feed-forward layer with the multi-head-attention mechanism layer based on the CNN+MF architecture. The optimal performance and the corresponding loss weight value of each model are shown in Table 2 and Figure 3. The results show that the CNN+MM model achieved the highest AUROC and AUPRC scores for both GAC_BM and AAC_BM datasets. The specific analysis is as follows:

First, comparing CNN+TF with the CNN, the AUROC scores of CNN+TF were 1.93% and 2.07% higher on AAC_BM and GAC_BM, respectively, and the AUPRC scores of CNN+TF were 1.53% and 1.72% higher than the values for the CNN. These findings highlight the ability of the CNN+TF model to capture deep semantics from RNA sequences, surpassing the performance of the CNN alone.

Additionally, a comparison was made between the CNN+MF and CNN+TF models. The AUROC scores of CNN+MF were 0.68% and 0.32% higher than those for the CNN+TF on AAC_BM and GAC_BM, respectively, and the AUPRC scores of CNN+MF were 0.92% and 0.34% higher on

AAC_BM and GAC_BM, respectively. This reason may be attributed to the mixed correlations between positional encoding and word embeddings in the CNN+TF model, which introduce unnecessary randomness to the attention mechanism and limit the model's expressiveness. Therefore, CNN+MF affords performance improvement, since the CNN+MF model directly connects the multi-head-attention mechanism layer without connecting the positional encoding layer.

Table 2. Performance of different models trained by using their respective best value λ in a 5-fold cross-validation test. Values in bold indicate the best performance.

Datasets	Classifiers	AUROC	ACC (%)	Sen (%)	Pre (%)	MCC (%)	Spe (%)	F-1 (%)	AUPRC	PCC	Time (s)
AAC_BM	CNN (λ = 1)	0.8507	77.64	78.58	77.13	55.40	76.69	77.85	0.8382	0.5844	1
	CNN+BiL (λ = 1)	0.8767	**79.97**	82.61	78.47	**60.10**	77.33	80.49	0.8607	0.6137	136
	CNN+TF (λ = 1)	0.8700	79.01	**84.65**	76.08	58.54	73.38	80.13	0.8535	0.5986	4
	CNN+MF (λ = 1)	0.8768	79.75	83.99	77.43	59.77	75.51	**80.58**	0.8627	0.6040	4
	CNN+MM (λ = 0.6)	**0.8793**	79.57	80.94	**79.06**	59.98	**78.56**	79.99	**0.8636**	**0.6140**	25
GAC_BM	CNN (λ = 0.4)	0.8506	76.95	77.24	76.80	54.66	76.66	77.02	0.8394	0.5923	1
	CNN+BiL (λ = 1)	0.8742	79.61	**85.46**	76.51	**59.71**	73.75	**80.74**	0.8595	**0.6199**	108
	CNN+TF (λ = 0.4)	0.8713	79.46	80.94	78.61	59.04	77.97	79.76	0.8566	0.6005	4
	CNN+MF (λ = 1)	0.8745	79.61	81.19	78.70	59.35	78.03	79.93	0.8600	0.6112	4
	CNN+MM (λ = 0.6)	**0.8772**	**79.06**	78.69	**79.28**	58.48	**79.43**	79.80	**0.8635**	0.6156	20

Times: Running time per epoch for model training

Third, the study compared the performance of the CNN+MM and CNN+MF models. The AUROC scores of CNN+MM were 0.25% and 0.27% higher than the CNN+MF on AAC_BM and GAC_BM, respectively, and the AUPRC scores of CNN+MM were 0.09% and 0.35% higher on AAC_BM and GAC_BM, respectively. This disparity could be attributed to the ineffective capture of contextual information by the feed-forward networks in the transformer structure, as feed-forward networks lack the ability to effectively capture global contextual information, resulting in a less accurate understanding of long-term dependencies or global patterns in the sequence. The CNN+MM model replaces the feed-forward layer with the multi-head-attention mechanism layer based on the CNN+MF architecture. This modification allows the model to capture more complex and fine-grained features within the input sequence.

Fourth, a comparison was made between the CNN+MM and CNN+BiL models. The AUROC scores of CNN+MM were 0.26% and 0.3% higher than those for CNN+BiL on AAC_BM and GAC_BM, respectively, while the AUPRC scores of CNN+MM were 0.29% and 0.39% higher on AAC_BM and GAC_BM, respectively. The improved performance of CNN+MM may be attributed

to the inclusion of multi-head-attention, which excels at capturing long-range dependencies and global contextual information. This allows the model to better understand relationships and important semantic connections within the input sequence. Furthermore, the experimental results demonstrate that CNN+BiL has a significantly longer training time per epoch than CNN+MM; particularly, it is five times longer than that of CNN+MM, likely due to the parallel computation of the CNN and multi-head-attention, which enables simultaneous processing of multiple input elements or subtasks. In contrast, BiLSTM's sequential nature, where computations are performed step-by-step, may result in slower computations than parallelizable operations.

Finally, in the case of the regression task, there is little difference between CNN+MM and CNN+BiL; specifically, the PCCs of CNN+MM and CNN+BiL differ by 0.03 and –0.43 on AAC_BM and GAC_BM, respectively. Interestingly, although the loss weight ratio between the classification task and the regression task of the CNN+MM model was 1:0.6, it had little effect on the correlation coefficient of the regression. The reason may be that the CNN+MM model has a more comprehensive and accurate understanding of the underlying data.

In summary, the CNN+MM classifier effectively captures sequence details on the AAC_BM and GAC_BM datasets, outperforming other models in terms of AUROC and AUPRC on the classification task.

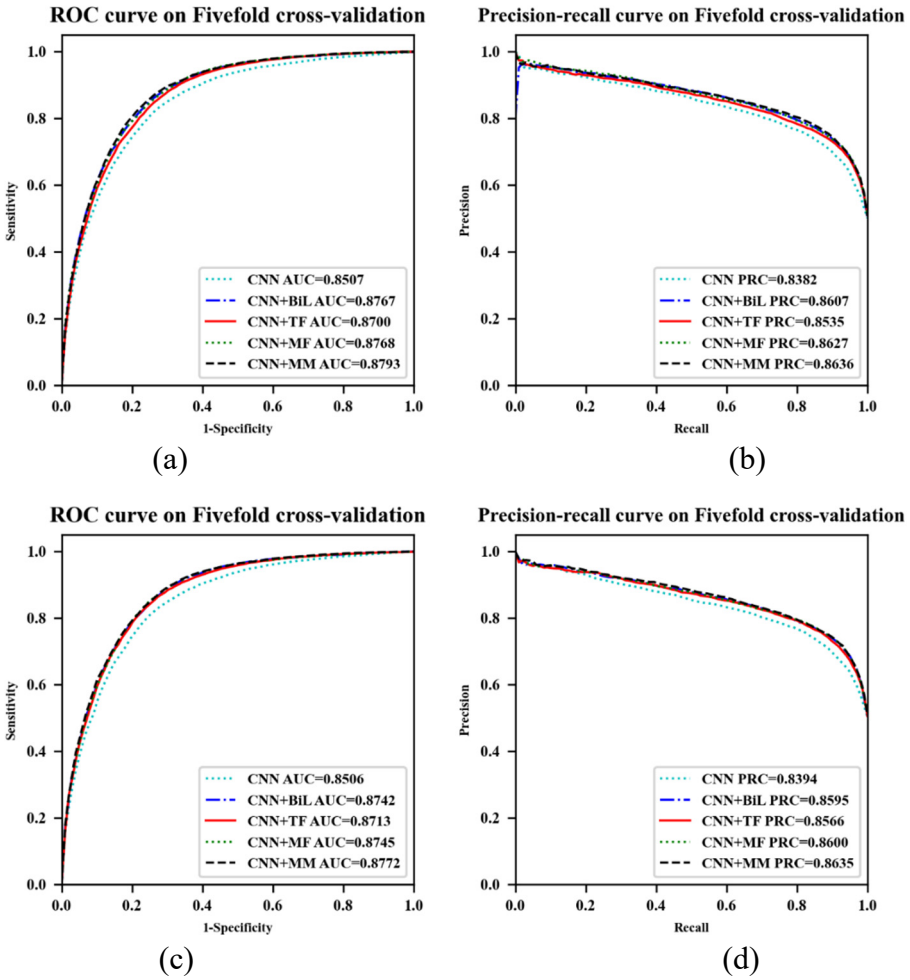

Figure 3. Performance of the different models through 5-fold cross-validation. (a) AUROC for AAC_BM, (b) AUPRC for AAC_BM, (c)AUROC for GAC_BM, (d) AUPRC for GAC_BM.

3.2. Comparison with other different learning models based on independent dataset

Furthermore, this section presents a comparison of prediction performance between different models in single-task and multi-task settings. The experimental setup involved encoding sequences using one-hot encoding and applying the CNN, CNN+BiL, CNN+TF, CNN+MF, and CNN+MM models to predict modification sites based on the independent dataset. The ratio of the classification loss to the regression loss affects the performance of the models, which is one of the hyper parameters. Only the training sets can be used while choosing the hyper parameters. Therefore, when evaluating the performance of various models based on independent test sets, the models must use the optimal loss weight value for each model as obtained from the training set; the values were obtained as shown in the second column of Table 2.

As shown in Figure 4 and Table 3, except for the CNN+BiL model, the AUROC and the AUPRC scores for the multi-task model based on the two datasets were better than those for the single-task model. In addition, the multi-task model can also calculate the PCC at the same time, so the multi-task model is more efficient than the single-task model.

The comparison results, demonstrate that CNN+MM, operating under the multi-tasking framework, outperforms other models across various evaluation metrics such as the AUROC, AUPRC, PCC, ACC, and MCC. Specifically, for AAC_IND and GAC_IND sites, CNN+MM achieved AUROC values of 0.8888 and 0.888, respectively, exhibiting better performance than other methods. In contrast, CNN+BiL did not incorporate the multi-head-attention mechanism, potentially limiting their ability to capture global contextual information as compared to CNN+MM. CNN+MF and CNN+TF contain the multi-head-attention layer. However, they both contain the feed-forward layer, which lacks the ability to effectively capture deeper global information compared to multi-head-attention layers, resulting in a less accurate understanding of long-term dependencies or global patterns in the sequence.

3.3. Comparison with other different learning models in the target domain-stage

Considering the similarity between the two-stage classification tasks, we chose to leverage the feature extraction layer of the source domain-stage model to construct the target domain-stage model. Specifically, in the transfer learning process, we initialize the parameters of the target domain-stage model with the feature extraction layer (excluding the output layer) and the corresponding weights of the source domain-stage model. During training, we optimize all of the weights of the target domain-stage model without freezing them.

To evaluate the effectiveness of transfer learning compared to training from scratch, we also trained the same network without using the weights obtained from the source domain-stage model. Furthermore, to assess the performance of multi-task transfer learning against single-task fine-tuning, we also trained the same network indirectly through single-task transfer learning. We conducted three sets of comparative experiments on the GAC_hr_BM datasets, and the results of these experiments are shown in Table 3.

Table 3. Evaluation results for the different models under single-task and multi-task conditions. Values in bold indicate the best performance.

Datasets	Task type	Classifiers	AUROC	ACC (%)	Sen (%)	Pre (%)	MCC (%)	Spe (%)	F-1 (%)	AUPRC	PCC
AAC_IND	Single-task	CNN	0.8579	78.63	81.03	77.32	57.32	76.23	79.13	0.8510	
		CNN+BiL	0.8829	80.04	79.90	80.13	60.09	80.19	80.01	0.8737	
		CNN+TF	0.8736	78.54	76.55	79.71	57.12	**80.52**	78.10	0.8642	
		CNN+MF	0.8763	78.97	77.72	79.72	57.96	80.23	78.71	0.8657	
		CNN+MM	0.8820	80.06	83.10	78.34	60.23	77.03	80.65	0.8724	
	Multi-task	CNN (λ = 1)	0.8581	78.55	81.1	77.17	57.18	76.01	79.09	0.8516	0.5946
		CNN+BiL (λ = 1)	0.8801	79.75	79.1	**80.15**	59.51	80.41	79.62	0.8682	0.6042
		CNN+TF (λ = 1)	0.8768	80.44	83.1	78.91	60.97	77.79	80.95	0.8666	0.613
		CNN+MF (λ = 1)	0.8816	80.12	**89.35**	75.42	61.29	70.88	**81.80**	0.8697	0.6053
		CNN+MM (λ = 0.6)	**0.8888**	**80.97**	83.72	79.36	**62.03**	78.23	81.48	**0.8775**	**0.6219**
GAC_IND	Single-task	CNN	0.8553	77.92	84.78	74.55	56.37	71.06	79.33	0.8500	
		CNN+BiL	0.8767	79.97	82.45	78.55	60.01	77.48	80.45	0.8667	
		CNN+TF	0.8699	77.26	92.94	70.75	57.41	61.58	80.34	0.8589	
		CNN+MF	0.8635	73.29	**95.67**	66.09	52.09	50.91	78.18	0.8509	
		CNN+MM	0.8761	80.20	84.23	77.94	60.59	76.16	80.96	0.8660	
	Multi-task	CNN (λ = 0.4)	0.8560	78.28	83.45	75.63	56.87	73.11	79.35	0.8507	0.5935
		CNN+BiL (λ = 1)	0.8729	78.87	79.63	78.45	57.75	78.12	79.03	0.8665	0.6124
		CNN+TF (λ = 0.4)	0.8734	79.58	86.69	75.90	59.77	72.47	80.94	0.8622	0.6029
		CNN+MF (λ = 1)	0.8795	80.20	83.59	78.28	60.53	76.80	80.85	0.8698	0.6202
		CNN+MM (λ = 0.6)	**0.8880**	**81.11**	83.23	**79.84**	**62.27**	**78.99**	**81.50**	**0.8783**	**0.6343**

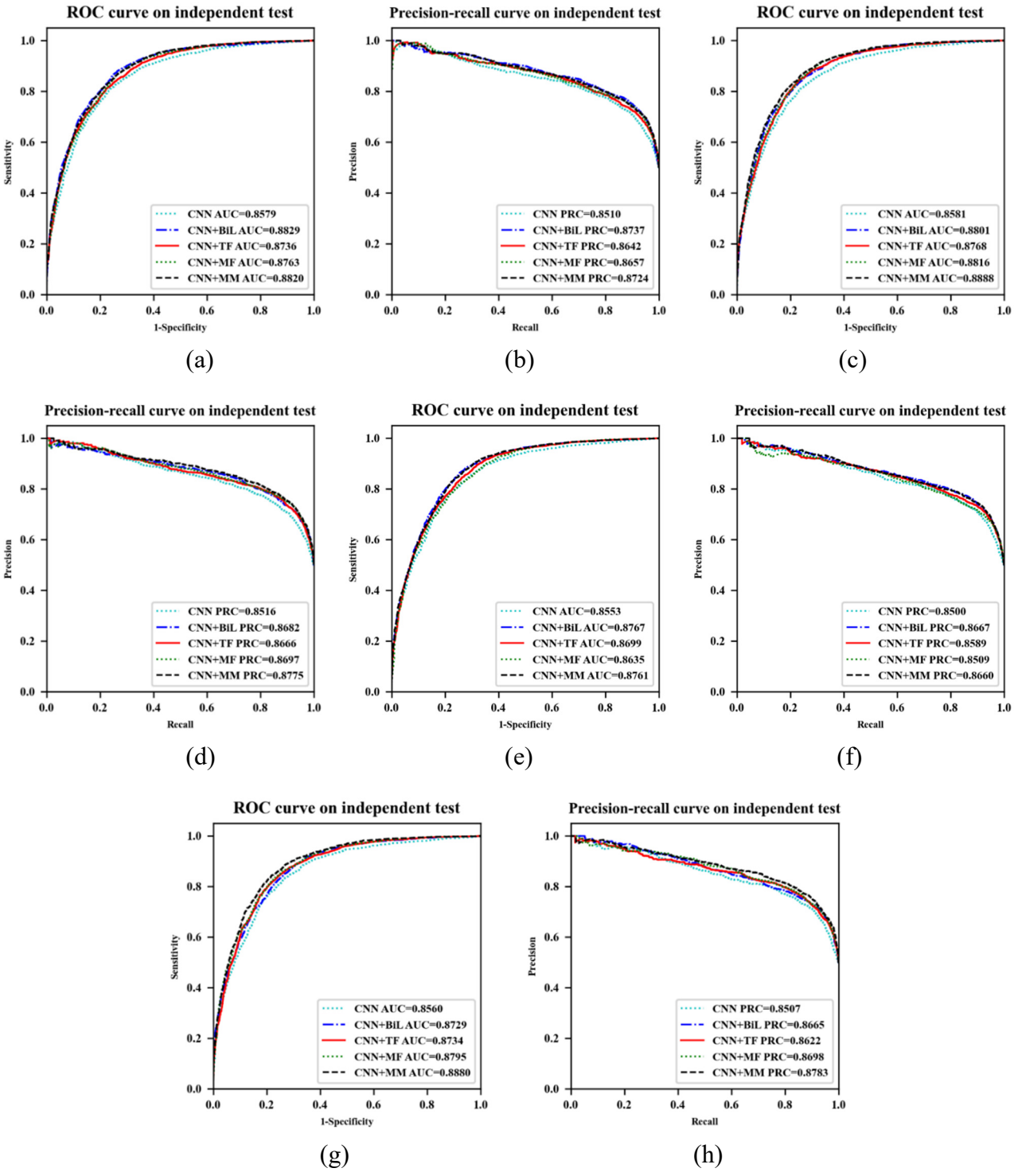

Figure 4. Performance results for the different models under single-task and multi-task conditions. (a) AUROC_single_task for AAC_IND, (b) AUPRC_single_task for AAC_IND, (c) AUROC_multi_task for AAC_IND, (d) AUPRC_multi_task for AAC_IND, (e) AUROC_single_task for GAC_IND, (f) AUPRC_single_task for GAC_IND, (g) AUROC_multi_task for GAC_IND, (h) AUPRC_multi_task for GAC_ IND.

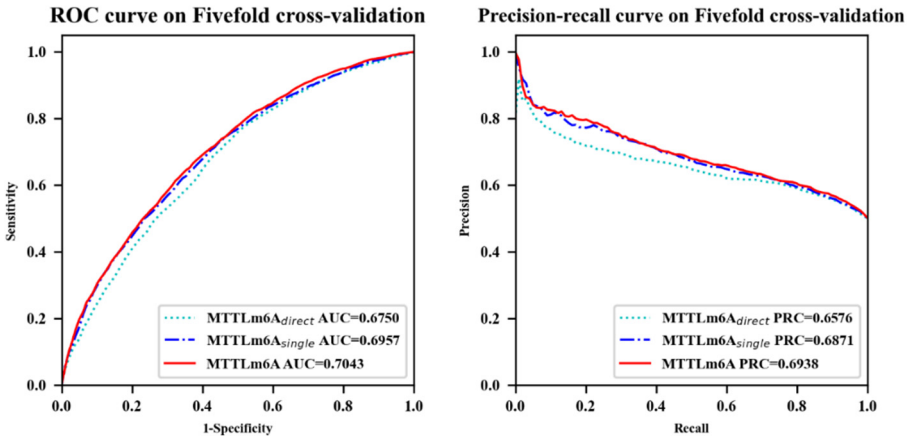

Figure 5. Performance results for the different models in terms 5-fold cross-validation based on the GAC_hr_BM dataset.

Table 4. Evaluation results for the different models from the perspective of 5-fold cross-validation based on the GAC_hr_BM dataset.

Classifiers	AUROC	ACC (%)	Sen (%)	Pre (%)	MCC (%)	Spe (%)	F-1 (%)	AUPRC
MTTLm^{6}A $_{direct}$	0.6750	62.33	61.69	62.49	24.81	62.97	62.09	0.6576
MTTLm^{6}A $_{single}$	0.6957	63.74	60.28	64.77	27.69	67.21	62.44	0.6871
MTTLm^{6}A	**0.7043**	**64.40**	**65.53**	64.08	**28.94**	63.26	**64.80**	**0.6938**

MTTLm$^6A_{direct}$ refers to the model trained directly without transfer learning; MTTLm$^6A_{single}$ indicates the model trained indirectly through single-task transfer learning.

As shown in Figure 5 and Table 4, the AUROC and AUPRC values for MTTLm^{6}A$_{single}$ were 2.07% and 2.92% higher than those for MTTLm^{6}A$_{direct}$. This improvement suggests that transfer learning enhances the model's performance relative to training without transfer learning. The reason for this improvement is that transfer learning allows the model to leverage knowledge gained from related tasks or domains, allowing it to generalize well to unseen data. Furthermore, the performance of the model trained through multi-task transfer learning was significantly better than that of the model trained through single-task transfer learning. Specifically, the AUROC and AUPRC values for MTTLm^{6}A were 0.86% and 0.68% higher than those for MTTLm^{6}A$_{single}$, respectively. This result can be attributed to the complementary information provided by the combination of classification and regression tasks. While the classification task focuses on predicting discrete class labels, the regression task aims to estimate continuous values. By jointly training the model on both tasks, MTTLm^{6}A can effectively utilize the complementary information from both tasks to improve its understanding of the data and make more accurate predictions.

3.4. Comparison with state-of-the-art approaches

Finally, MTTLm^{6}A was compared with other state-of-the-art approaches on the GAC_hr_IND datasets, including m^{6}A-word2vec, MultiRM, and MTDeepM6A-2S. To make the comparison more convincing, we included the MTTLm^{6}A$_{single}$ model in the evaluation.

As shown in Figure 6 and Table 5, the AUROC and AUPRC values for MTTLm^{6}A were higher

than those obtained for the other approaches. In particular, compared to MTDeepM^{6}A-2S, the second-best performing model utilizing multi-task transfer learning, MTTLm^{6}A, demonstrated an improvement of 0.37% in AUROC and 0.58% in AUPRC. This enhancement can be attributed to MTTLm^{6}A's ability to capture long-range dependencies and global contextual information in the input sequence compared to MTDeepM^{6}A-2S.

Furthermore, the AUROC and AUPRC values for MTTLm^{6}A were 1.14% and 0.37% higher than those for MTTLm^{6}A$_{single}$, respectively. The incorporation of multi-task learning in the MTTLm^{6}A model allows for joint training on both classification and regression tasks. This integration allows the model to learn shared representations and extract more informative features that benefit both tasks. By collectively optimizing these tasks, the model achieves improved overall performance. In contrast, MTTLm^{6}A$_{single}$ focuses solely on the classification task, potentially limiting its ability to capture the full accuracy of the data.

Additionally, MTTLm^{6}A surpassed MultiRM by a notable margin, exhibiting AUROC and AUPRC improvements of 7.66% and 8.24%, respectively. This highlights the effectiveness of MTTLm^{6}A in addressing the challenges posed by small sample classification data modeling problems.

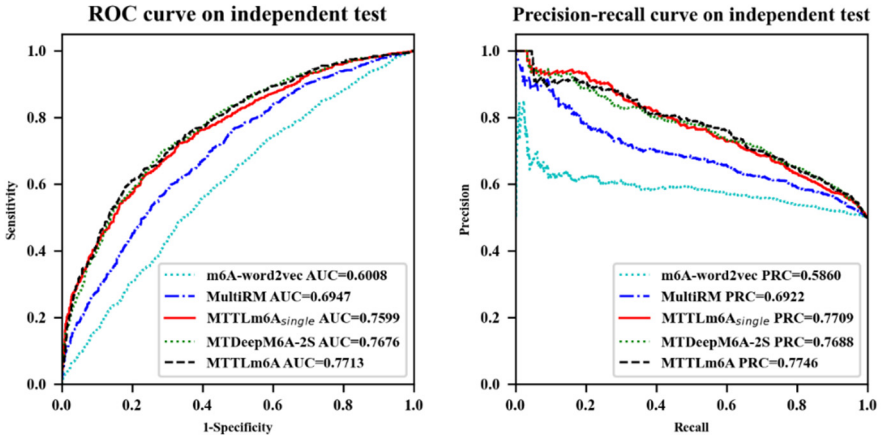

Figure 6. Performance of MTTLm^{6}A and other models based on the GAC_hr_IND dataset.

Table 5. Comparison of the performance of different models based on the GAC_hr_IND dataset.

Classifiers	AUROC	ACC (%)	Sen (%)	Precision (%)	MCC (%)	Spe (%)	F-1 (%)	AUPRC
m^{6}A-word2vec	0.6008	57.78	58.96	57.60	15.57	48.83	58.27	0.586
MultiRM	0.6947	63.75	69.08	62.43	27.66	44.67	65.59	0.6922
MTTLm^{6}A$_{single}$	0.7599	69.10	67.56	69.71	38.23	70.65	68.62	0.7709
MTDeepM6A-2S	0.7676	**70.44**	66.81	**72.04**	**40.98**	**74.07**	69.32	0.7688
MTTLm^{6}A	**0.7713**	69.85	**74.81**	68.06	39.90	64.89	**71.28**	**0.7746**

In order to evaluate the reliability of the model, the m6A-word2vec, MultiRM, MTTLm^{6}Asingle, MTDeepM6A-2S, and MTTLm^{6}A models were applied for 100 replicate experiments on the same independent test sets of m^{6}A, respectively. After 100 replicates, we tested the statistical significance of AUROC values between different tools by using the student's t-test [70]. The results are as shown in Table 6.

Table 6. Statistically significant correlation matrix for the difference in the performance between the five selected classifiers.

Modification type	Classifiers	Classifiers				
		m6A-word2vec	MultiRM	MTTLm^{6}A$_{singl}$	MTDeepM6A-2S	MTTLm^{6}A
m^{6}A	m6A-word2vec					
	MultiRM	0				
	MTTLm^{6}A$_{single}$	0	0			
	MTDeepM6A-2S	0	0	0		
	MTTLm^{6}A	0	0	0	0	

3.5. Assessing model generalization ability

In order to evaluate the generalization ability of the MTTLm^{6}A model, MTTLm^{6}A was compared and analyzed against m6A-word2vec, MultiRM, MTTLm^{6}Asingle and MTDeepM6A-2S by using the training set and independent set of m^{1}A sites of Homo sapiens. As shown in Table 7, the AUROC and AUPRC values for MTTLm^{6}A were higher than those obtained for other approaches. In particular, compared to MTTLm^{6}A$_{single}$, the second-best performing model, MTTLm^{6}A demonstrated an improvement of 1.47% in AUROC and 0.63% in AUPRC. This shows that multi-task learning is effective in improving model performance. In summary, the MTTLm^{6}A model has good versatility in predicting different methylation sites in different species.

Table 7. Comparison of the performance of different models based on the hr_IND dataset.

Classifiers	AUROC	ACC (%)	Sen (%)	Precision (%)	MCC (%)	Spe (%)	F-1 (%)	AUPRC
m6A-word2vec	0.9095	**91.67**	**100.00**	**85.71**	**84.52**	83.33	**92.31**	0.8722
MultiRM	0.9126	91.23	99.12	85.61	83.50	83.33	91.87	0.8818
MTTLm^{6}A$_{single}$	0.9143	90.79	98.25	85.50	82.50	83.33	91.43	0.8841
MTDeepM6A-2S	0.894	91.23	99.12	85.61	83.50	83.33	91.87	0.8509
MTTLm^{6}A	**0.929**	91.23	99.12	85.61	83.50	**83.33**	91.87	**0.8904**

3.6. Web server

We have developed a user-friendly web server, accessible at http://47.242.23.141/MTTLm6A/index.php, to facilitate the utilization of the MTTLm^{6}A model as a tool for predicting the base-resolution m^{6}A sites. Simply type or paste the RNA sequence of interest into the designated input area. To receive the prediction results, kindly provide your email address in the corresponding box and click the "submit" button. After a brief calculation period, the prediction results will be presented in a clear and organized table format. This intuitive web server provides researchers with an efficient and convenient platform to leverage the MTTLm^{6}A model to quickly predict the base-resolution m^{6}A site.

4. Discussion

To assess the impact of different loss weights on the source domain-stage model, we conducted an optimization process by using the grid search method with AAC_BM and GAC_BM datasets. Through this process, we identified an optimal weight ratio of 1:0.6. The evaluation metrics, as displayed in Table 8, validate the effectiveness of this weight configuration. This finding aligns with the conclusions drawn by Kendall et al. [71], who emphasized the importance of relative weighting in multi-task learning scenarios. Their research demonstrated that numerous DL applications benefit from incorporating multiple regression and classification objectives, but the performance of such systems relies heavily on the appropriate weighting assigned to each task's loss function. By determining the optimal loss weight ratio for our source domain-stage model, we aimed to enhance its predictive capabilities and ensure a balanced influence of the classification and regression tasks. This optimization process allows us to fully leverage the benefits of multi-task learning and maximize the performance of our model.

Table 8. Performance of models trained by using different loss weights in a 5-fold cross-validation test. Values in bold indicate the best performance.

Datasets	Weight ratio	AUROC	ACC (%)	Sen (%)	Pre (%)	MCC (%)	Spe (%)	F-1 (%)	AUPRC	PCC
AAC_BM	1:0.1	0.8753	79.3	80.59	78.56	59.05	78.01	79.56	0.8579	0.5894
	1:0.2	0.8745	77.95	74.85	79.80	56.9	81.05	77.25	0.8586	0.6019
	1:0.3	0.8788	78.74	77.67	79.36	58.38	79.81	78.51	0.8615	0.6093
	1:0.4	0.8734	76.56	70.76	80.05	54.77	**82.36**	75.12	0.8574	0.6126
	1:0.5	0.8745	78.16	77.85	78.34	57.62	78.48	78.10	0.8590	0.6078
	1:0.6	**0.8793**	**79.57**	**80.94**	79.06	**59.98**	78.56	**79.99**	**0.8636**	0.614
	1:0.7	0.8754	77.6	73.8	**79.86**	56.65	81.39	76.71	0.8587	0.6108
	1:0.8	0.8753	78.04	76.7	78.81	56.98	79.38	77.74	0.8592	**0.6218**
	1:0.9	0.8749	77.85	75.95	78.94	56.95	79.74	77.42	0.8584	0.6124
	1:1.0	0.8756	78.66	77.96	79.06	58.11	79.36	78.51	0.8595	0.5996
GAC_BM	1:0.1	0.8746	79.62	81.97	78.30	**59.59**	77.28	80.09	0.8584	0.6011
	1:0.2	0.8771	**79.65**	80.76	78.88	59.54	77.54	**80.29**	0.8612	0.6033
	1:0.3	0.8761	79.58	80.27	79.18	59.56	78.9	79.72	0.8627	0.6095
	1:0.4	0.8764	79.21	**83.78**	76.77	59.31	74.65	80.12	0.8634	0.6144
	1:0.5	0.8746	78.62	82.57	76.52	58.25	74.67	79.43	0.8605	0.6136
	1:0.6	**0.8772**	79.06	78.69	79.28	58.48	79.43	79.80	**0.8635**	0.6156
	1:0.7	0.8733	78.8	77.89	**79.33**	57.94	**79.71**	78.61	0.8584	0.6187
	1:0.8	0.8753	79.1	81.77	77.63	58.93	76.44	79.65	0.8589	0.6150
	1:0.9	0.8766	79.60	80.99	78.8	59.39	78.21	79.88	0.8613	0.6218
	1:1.0	0.8756	78.66	77.96	79.06	58.11	79.36	78.51	0.8595	**0.6220**

To evaluate the effectiveness of our multi-task learning architecture, we conducted two separate experiments: single-task learning for the classification task and single-task learning for the regression task. Both experiments used the CNN+MM network. Table 9 presents the cross-validation results of the single-task classification model and the multi-task model based on two different benchmark datasets. Our findings reveal that, in the case of multi-task learning, the AUROC values for the

classification task were 0.8794 and 0.8772 for AAC_BM and GAC_BM, respectively. In comparison, the AUROC values for the classification task in single-task learning were 0.8774 and 0.8769 for AAC_BM and GAC_BM, respectively. Therefore, the performance of the multi-task classification model surpassed that of the single-task classification model for both AAC_BM and GAC_BM. The reason may be that multiple related tasks help to regularize each other and a more robust representation can be learned; thus, multi-task learning is usually believed to improve network performance. These results align with a study by Ruder [72], which emphasizes that multi-task learning allows the model to focus its attention on relevant features, as other tasks provide additional evidence for determining the relevance or irrelevance of those features. By adopting a multi-task learning approach, our model benefits from the shared representation and complementary information across tasks, leading to improved classification performance. Specifically, the model adds NSES information, which helps identify poor-quality methylation sites. These findings underscore the effectiveness of our multi-task learning architecture in enhancing model performance and feature relevance assessment.

Table 9. Performance comparison between the single-task classification models and the multi-task classification models based on AAC_BM, and GAC_BM, respectively. Values in bold indicate the best performance.

Datasets	task (Weight-ratio)	AUROC	ACC (%)	Sen (%)	Pre (%)	MCC (%)	Spe (%)	F-1 (%)	AUPRC
AAC_BM	single-task	0.8774	78	74.71	**79.97**	57.33	**81.29**	77.25	0.8613
	multi-task (1:0.6)	**0.8794**	**79.57**	**80.94**	79.06	**59.98**	78.56	**79.99**	**0.8636**
GAC_BM	single-task	0.8769	**79.76**	**79.61**	**79.85**	**59.7**	**79.91**	79.73	0.8613
	multi-task (1:0.6)	**0.8772**	79.06	78.69	79.28	58.48	79.43	**79.80**	**0.8635**

Table 10 presents the PCC results for the cross-validation of the single-task regression model and the multi-task model on AAC_BM and GAC_BM. The results indicate that the multi-task model slightly outperforms the single-task regression model on both datasets. Specifically, the correlation coefficients for the regression task were 0.614 and 0.6156 for multi-task learning, while they were 0.6042 and 0.6 for single-task learning for AAC_BM and GAC_BM, respectively. Interestingly, despite the loss weight ratio between the classification and regression tasks of the CNN+MM model in the source domain-stage being 1:0.6, the impact on the correlation coefficients of the regressions is minimal. This suggests that the improved performance of our multi-task model relative to the single-task regression model can be attributed to several factors, including the utilization of complementary information, shared feature representation, regularization techniques, and knowledge transfer between tasks. By leveraging multi-task learning, our model benefits from the synergistic effects of multiple tasks, leading to enhanced PCCs. This highlights the advantages of incorporating related tasks and sharing representations, ultimately resulting in a more comprehensive and accurate understanding of the underlying data. By considering the interplay between tasks and the complementary nature of their information, we can leverage multi-task learning to further improve the performance of our model and achieve superior results across a range of metrics.

Table 10. PCC results for cross-validation of single-task regression model and multi-task model on AAC_BM and GAC_BM. Values in bold indicate the best performance.

Datasets	PCC for multi-task regression models	PCC for single-task regression models
AAC_BM	**0.614**	0.6042
GAC_BM	**0.6156**	0.6

In summary, our results demonstrate that the multi-task learning approach outperforms single-task learning on all four tasks on both the AAC_BM and GAC_BM datasets. Furthermore, the multi-task learning framework offers an added advantage of increased efficiency compared to the single-task model. By simultaneously addressing multiple tasks, the model can accomplish more with fewer resources and reduced computational overhead. In conclusion, our findings highlight the superiority of the multi-task model over the single-task model.

5. Conclusions

The contribution of this paper lies in the development of a novel predictor called MTTLm^{6}A, which utilizes a multi-task learning and transfer learning approach based on an improved transformer architecture to identify base-resolution mRNA m^{6}A sites. Experimental results on Saccharomyces cerevisiae m^{6}A and Homo sapiens m^{1}A data demonstrate that MTTLm^{6}A respectively achieved AUROC values of 77.13% and 92.9%, outperforming the state-of-the-art models. At the same time, it shows that the model has strong generalization ability. But the model has a limitation, which is that source domain-stage training requires samples with NSES information, which is a necessary condition for multi-task learning.

Furthermore, considering that multi-task learning tends to benefit from an increasing number of tasks, we intend to delve deeper into the characteristics of methylation parameters. By incorporating more effective tasks into the learning framework, we will extract more base-resolution methylation sequences from low-resolution methylation sequences of different species by using the MTTLm^{6}A model for scientific research.

In conclusion, the development of MTTLm^{6}A, its promising performance, and future research directions contribute to the advancement of computational methods for the identification of methylation sites, demonstrating its potential for broader application and further refinement in the future.

Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (31871337 and 61971422), and the “333 Project” of Jiangsu (BRA2020328).

Conflict of interest

The authors declare no conflict of interest.

References

1. A. Nossent, The epitranscriptome: RNA modifications in vascular remodelling, *Atherosclerosis*, **374** (2023), 24–33. https://doi.org/10.1016/j.atherosclerosis.2022.11.004
2. H. H. Shi, P. W. Chai, R. B. Jia, X. Q. Fan, Novel insight into the regulatory roles of diverse RNA modifications: Re-defining the bridge between transcription and translation, *Mol. Cancer*, **19** (2020), 1–17. https://doi.org/10.1186/s12943-020-01194-6
3. S. Ramasamy, S. Mishra, S. Sharma, S. S. Parimalam, T. Vaijayanthi, Y. Fujita, et al., An informatics approach to distinguish RNA modifications in nanopore direct RNA sequencing, *Genomics*, **114** (2022), 1–8. https://doi.org/10.1016/j.ygeno.2022.110372
4. S. H. Boo, Y. K. Kim, The emerging role of RNA modifications in the regulation of mRNA stability, *Exp. Mol. Med.*, **52** (2020), 400–408. https://doi.org/10.1038/s12276-020-0407-z
5. L. Cui, R. Ma, J. Cai, C. Guo, Z. Chen, L. Yao, et al., RNA modifications: Importance in immune cell biology and related diseases, *Signal Transduction Targeted Ther.*, **7** (2022), 1–26. https://doi.org/10.1038/s41392-022-01175-9
6. I. Orsolic, A. Carrier, M. Esteller, Genetic and epigenetic defects of the RNA modification machinery in cancer, *Trends Genet.*, **39** (2023), 74–88. https://doi.org/10.1016/j.tig.2022.10.004
7. X. Bao, Y. Zhang, H. Li, Y. Teng, L. Ma, Z. Chen, et al., RM2Target: A comprehensive database for targets of writers, erasers and readers of RNA modifications, *Nucleic Acids Res.*, **51** (2023), 269–279. https://doi.org/10.1093/nar/gkac945
8. Y. Yan, J. Peng, Q. Liang, X. Ren, Y. Cai, B. Peng, et al., Dynamic m^6A-ncRNAs association and their impact on cancer pathogenesis, immune regulation and therapeutic response, *Genes Dis.*, **10** (2023), 135–150. https://doi.org/10.1016/j.gendis.2021.10.004
9. S. Nag, B. Goswami, S. D. Mandal, P. S. Ray, Cooperation and competition by RNA-binding proteins in cancer, *Semin. Cancer Biol.*, **86** (2022), 286–297. https://doi.org/10.1016/j.semcancer.2022.02.023
10. J. W. Wenger, K. Schwartz, G. Sherlock, Bulk segregant analysis by high-throughput sequencing reveals a novel xylose utilization gene from saccharomyces cerevisiae, *Plos Genet.*, **6** (2010), 1–17. https://doi.org/10.1371/journal.pgen.1000942
11. M. J. Wakefield, Genomics—from Neanderthals to high-throughput sequencing, *Genome Biol.*, **7** (2006), 1–3. https://doi.org/10.1186/gb-2006-7-8-326
12. J. Hamfjord, A. M. Stangeland, T. Hughes, M. L. Skrede, K. M. Tveit, T. Ikdahl, et al., Differential expression of miRNAs in colorectal cancer: Comparison of paired tumor tissue and adjacent normal mucosa using high-throughput sequencing, *Plos One*, **7** (2012), 1–9. https://doi.org/10.1371/journal.pone.0034150
13. F. Ahmed, P. X. Zhao, A comprehensive analysis of isomirs and their targets using high-throughput sequencing data for Arabidopsis thaliana, *J. Nat. Sci. Biol. Med.*, **2** (2011), 1414–1429.
14. Y. Wang, A. Li, L. Zhang, M. Waqas, K. Mehmood, M. Iqbal, et al., Probiotic potential of Lactobacillus on the intestinal microflora against Escherichia coli induced mice model through high-throughput sequencing, *Microb. Pathogenesis*, **137** (2019), 1–9. https://doi.org/10.1016/j.micpath.2019.04.020
15. Z. Zhang, L. Q. Chen, Y. L. Zhao, C. G. Yang, I. A. Roundtree, Z. Zhang, et al., Single-base mapping of m(6)A by an antibody-independent method, *Sci. Adv.*, **5** (2019), 1–12. https://doi.org/10.1126/sciadv.aax0250

16. B. Linder, A. V. Grozhik, A. O. Olarerin-George, C. Meydan, C. E. Mason, S. R. Jaffrey, Single-nucleotide-resolution mapping of m^6A and m^6Am throughout the transcriptome, *Nat. Methods*, **12** (2015), 1–8. https://doi.org/10.1038/nmeth.3453
17. J. S. Abebe, R. Verstraten, D. P. Depledge, Nanopore-based detection of viral RNA modifications, *Mbio*, **13** (2022), 1–15. https://doi.org/10.1128/mbio.03702-21
18. M. Ramezanpour, S. S. W. Leung, K. H. Delgado-Magnero, B. Y. M. Bashe, J. Thewalt, Tieleman DP: Computational and experimental approaches for investigating nanoparticle-based drug delivery systems, *Bba-Biomembranes*, **1858** (2016), 1688–1709. https://doi.org/10.1016/j.bbamem.2016.02.028
19. S. Albaradei, M. Thafar, A. Alsaedi, C. V. Neste, X. Gao, Machine learning and deep learning methods that use omics data for metastasis prediction, *Comput. Struct. Biotechnol. J.*, **1** (2021), 5008–5018. https://doi.org/10.1016/j.csbj.2021.09.001
20. R. P. Bonidia, L. D. H. Sampaio, D. S. Domingues, A. R. Paschoal, F. M. Lopes, A. de Carvalho, et al., Feature extraction approaches for biological sequences: A comparative study of mathematical features, *Brief Bioinf.*, **22** (2021), 1–42. https://doi.org/10.1093/bib/bbab011
21. R. Wang, Y. Jiang, J. Jin, C. Yin, H. Yu, F. Wang, et al., DeepBIO: An automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis, *Nucleic Acids Res.*, **51** (2023), 3017–3029. https://doi.org/10.1093/nar/gkad055
22. W. S. Noble, What is a support vector machine?, *Nat. Biotechnol.*, **2006** (2006), 1565–1567. https://doi.org/10.1038/nbt1206-1565
23. M. A. Hall, *Correlation-Based Feature Selection for Machine Learning*, Ph.D thesis, The University of Waikato, 1999.
24. H, Motoda, H. Liu, Feature selection, extraction and construction, *Commun. IICM*, **5** (2002), 2.
25. H. Iuchi, T. Matsutani, K. Yamada, N. Iwano, S. Sumi, S. Hosoda, et al., Representation learning applications in biological sequence analysis, *Comput. Struct. Biotechnol. J.*, **19** (2021), 3198–3208. https://doi.org/10.1016/j.csbj.2021.05.039
26. H. L. Li, Y. H. Pang, B. Liu, BioSeq-BLM: A platform for analyzing DNA, RNA and protein sequences based on biological language models, *Nucleic Acids Res.*, **49** (2021), 1–17. https://doi.org/10.1093/nar/gkaa1112
27. M. Leinonen, L. Salmela, Extraction of long k-mers using spaced seeds, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **19** (2022), 3444–3455. https://doi.org/10.1109/TCBB.2021.3113131
28. N. Ferruz, M. Heinzinger, M. Akdel, A. Goncearenco, L. Naef, C. Dallago, From sequence to function through structure: Deep learning for protein design, *Comput. Struct. Biotechnol. J.*, **21** (2023), 238–250. https://doi.org/10.1016/j.csbj.2022.11.014
29. D. Ofer, N. Brandes, M. Linial, The language of proteins: NLP, machine learning & protein sequences, *Comput. Struct. Biotechnol. J.*, **19** (2021), 1750–1758. https://doi.org/10.1016/j.csbj.2021.03.022
30. C. H. Yu, W. Chen, Y. H. Chiang, K. Guo, Z. M. Moldes, D. L. Kaplan, et al., End-to-end deep learning model to predict and design secondary structure content of structural proteins, *ACS Biomater. Sci. Eng.*, **8** (2022), 1156–1165. https://doi.org/10.1021/acsbiomaterials.1c01343
31. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, et al., Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions, *J. Big Data-Ger.*, **8** (2021), 1–74. https://doi.org/10.1186/s40537-020-00387-6

32. L. Zhang, G. S. Li, X. Y. Li, H. L. Wang, S. T. Chen, H. Liu, EDLm(6)APred: Ensemble deep learning approach for mRNA m(6)A site prediction, *BMC Bioinf.*, **22** (2021), 1–15. https://doi.org/10.1186/s12859-020-03881-z
33. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, preprint, arXiv: 1301.3781. https://doi.org/10.48550/arXiv.1301.3781
34. F. Wu, R. T. Yang, C. J. Zhang, L. N. Zhang, A deep learning framework combined with word embedding to identify DNA replication origins, *Sci. Rep. UK*, **11** (2021), 1–19. https://doi.org/10.1038/s41598-020-79139-8
35. S. Okada, M. Ohzeki, S. Taguchi, Efficient partition of integer optimization problems with one-hot encoding, *Sci. Rep. UK*, **9** (2019), 1–12. https://doi.org/10.1038/s41598-018-37186-2
36. F. Weninger, J. Bergmann, B. Schuller, Introducing CURRENNT: The munich open-source CUDA RecurREnt neural network toolkit, *J. Mach. Learn. Res.*, **16** (2015), 547–551.
37. H. L. Wang, H. Liu, T. Huang, G. S. Li, L. Zhang, Y. J. Sun, EMDLP: Ensemble multiscale deep learning model for RNA methylation site prediction, *BMC Bioinf.*, **23** (2022), 1–22. https://doi.org/10.1186/s12859-021-04477-x
38. Y. Su, A parallel computing and mathematical method optimization of CNN network convolution, *Microprocess Microsy*, **80** (2021), 1–7. https://doi.org/10.1016/j.micpro.2020.103571
39. K. Ma, C. H. Tang, W. J. Zhang, B. K. Cui, K. Ji, Z. X. Chen, et al., DC-CNN: Dual-channel convolutional neural networks with attention-pooling for fake news detection, *Appl. Intell.*, **53** (2023), 8354–8369. https://doi.org/10.1007/s10489-022-03910-9
40. M. Tahir, M. Hayat, K. T. Chong, Prediction of N6-methyladenosine sites using convolution neural network model based on distributed feature representations, *Neural Networks*, **129** (2020), 385–391. https://doi.org/10.1016/j.neunet.2020.05.027
41. Z. Chen, P. Zhao, F. Y. Li, Y. N. Wang, A. I. Smith, G. I. Webb, et al., Comprehensive review and assessment of computational methods for predicting RNA post-transcriptional modification sites from RNA sequences, *Briefings Bioinf.*, **21** (2020), 1676–1696. https://doi.org/10.1093/bib/bbz112
42. Y. Huang, N. N. He, Y. Chen, Z. Chen, L. Li, BERMP: A cross-species classifier for predicting m(6)A sites by integrating a deep learning algorithm and a random forest approach, *Int. J. Biol. Sci.*, **14** (2018), 1669–1677. https://doi.org/10.7150/ijbs.27819
43. Z. Chen, P. Zhao, F. Y. Li, T. T. Marquez-Lago, A. Leier, J. Revote, et al., iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data, *Briefings Bioinf.*, **21** (2020), 1047–1057. https://doi.org/10.1093/bib/bbz041
44. M. Riva, P. Gori, F. Yger, I. Bloch, Is the U-NET directional-relationship aware?, in *2022 IEEE International Conference on Image Processing (ICIP)*, (2022), 1–5. https://doi.org/10.1109/ICIP46576.2022.9897715
45. Q. H. Vo, H. T. Nguyen, B. Le, M. L. Nguyen, Multi-channel LSTM-CNN model for Vietnamese sentiment analysis, in *2017 9th International Conference on Knowledge and Systems Engineering*, (2017), 24–29. https://doi.org/10.1109/KSE.2017.8119429
46. Y. Q. Zhang, M. Hamada, DeepM6ASeq: Prediction and characterization of m6A-containing sequences using deep learning, *BMC Bioinf.*, **19** (2018), 1–11. https://doi.org/10.1186/s12859-017-2006-0

47. T. Song, X. D. Zhang, M. Ding, A. Rodriguez-Paton, S. D. Wang, G. Wang, DeepFusion: A deep learning based multi-scale feature fusion method for predicting drug-target interactions, *Methods*, **204** (2022), 269–277. https://doi.org/10.1016/j.ymeth.2022.02.007
48. Z. T. Song, D. Y. Huang, B. W. Song, K. Q. Chen, Y. Y. Song, G. Liu, et al., Attention-based multi-label neural networks for integrated prediction and interpretation of twelve widely occurring RNA modifications, *Nat. Commun.*, **12** (2021), 1–11. https://doi.org/10.1038/s41467-020-20314-w
49. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, preprint, arXiv: 1409.0473. https://doi.org/10.48550/arXiv.1409.0473
50. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, *Adv. Neural Inf. Process. Syst.*, **30** (2017), 1–15.
51. T. Shen, J. Jiang, T. Y. Zhou, S. R. Pan, G. D. Long, C. Q. Zhang, DiSAN: Directional self-attention network for RNN/CNN-free language understanding, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2018), 5446–5455. https://doi.org/10.1609/aaai.v32i1.11941
52. Y. Zhang, F. Ge, F. Li, X. Yang, J. Song, D. J. Yu, Prediction of multiple types of RNA modifications via biological language model, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **2023** (2023), 3205–3214. https://doi.org/10.1109/TCBB.2023.3283985
53. H. Shi, S. Li, X. Su, Plant6mA: A predictor for predicting N6-methyladenine sites with lightweight structure in plant genomes, *Methods*, **204** (2022), 126–131. https://doi.org/10.1016/j.ymeth.2022.02.009
54. P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, in *2018 Conference of the North American Chapter of the Association for Computational Linguistics*, (2018), 464–468. https://doi.org/10.18653/v1/N18-2074
55. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, et al., Exploring the limits of transfer learning with a unified text-to-text transformer, *J. Mach. Learn. Res.*, **21** (2020), 1–67.
56. G. Ke, D. He, T. Y. Liu, Rethinking the positional encoding in language pre-training, in *International Conference on Learning Representations 2021*, (2021), 1–14.
57. W. Chen, H. Tran, Z. Liang, H. Lin, L. Zhang, Identification and analysis of the N(6)-methyladenosine in the Saccharomyces cerevisiae transcriptome, *Sci. Rep.*, **5** (2015), 1–8. https://doi.org/10.1038/srep13859
58. W. Chen, H. Tang, H. Lin: MethyRNA, A web server for identification of N(6)-methyladenosine sites, *J. Biomol. Struct. Dyn.*, **35** (2017), 683–687. https://doi.org/10.1080/07391102.2016.1157761
59. R. G. Govindaraj, S. Subramaniyam, B. Manavalan, Extremely-randomized-tree-based prediction of N(6)-methyladenosine sites in saccharomyces cerevisiae, *Curr. Genomics*, **21** (2020), 26–33. https://doi.org/10.2174/1389202921666200219125625
60. L. Y. Wei, H. R. Chen, R. Su, M6APred-EL: A sequence-based predictor for identifying N6-methyladenosine sites using ensemble learning, *Mol. Ther-Nucl. Acids*, **12** (2018), 635–644. https://doi.org/10.1016/j.omtn.2018.07.004
61. W. Chen, H. Ding, X. Zhou, H. Lin, K. C. Chou, iRNA(m6A)-PseDNC: Identifying N-6-methyladenosine sites using pseudo dinucleotide composition, *Anal. Biochem.*, **561** (2018), 59–65. https://doi.org/10.1016/j.ab.2018.09.002

62. Y. Song, Y. Wang, X. Wang, D. Huang, A. Nguyen, J. Meng, Multi-task adaptive pooling enabled synergetic learning of RNA modification across tissue, type and species from low-resolution epitranscriptomes, *Briefings Bioinf.*, **24** (2023), 1–12. https://doi.org/10.1093/bib/bbad105
63. W. J. Sun, J. H. Li, S. Liu, J. Wu, H. Zhou, L. H. Qu, et al., RMBase: A resource for decoding the landscape of RNA modifications from high-throughput sequencing data, *Nucleic Acids Res.*, **44** (2016), 1–7. https://doi.org/10.1093/nar/gkw472
64. J. J. Xuan, W. J. Sun, P. H. Lin, K. R. Zhou, S. Liu, L. L. Zheng, et al., RMBase v2.0: Deciphering the map of RNA modifications from epitranscriptome sequencing data, *Nucleic Acids Res.*, **46** (2018), 327–334. https://doi.org/10.1093/nar/gkx934
65. Y. Tang, K. Chen, B. Song, J. Ma, X. Wu, Q. Xu, et al., M6A-Atlas: A comprehensive knowledgebase for unraveling the N6-methyladenosine (m6A) epitranscriptome, *Nucleic Acids Res.*, **49** (2021), 134–143. https://doi.org/10.1093/nar/gkaa692
66. D. Huang, B. Song, J. Wei, J. Su, F. Coenen, J. Meng, Weakly supervised learning of RNA modifications from low-resolution epitranscriptome data, *Bioinformatics*, **37** (2021), i222–i230. ttps://doi.org/10.1093/bioinformatics/btab278
67. H. Wang, S. H. Zhao, Y. C. Cheng, S. D. Bi, X. L. Zhu, MTDeepM6A-2S: A two-stage multi-task deep learning method for predicting RNA N6-methyladenosine sites of saccharomyces cerevisiae, *Front. Microbiol.*, **13** (2022), 1–14. https://doi.org/10.3389/fmicb.2022.999506
68. L. Fu, B. Niu, Z. Zhu, S. Wu, W. Li, CD-HIT: Accelerated for clustering the next-generation sequencing data, *Bioinformatics*, **28** (2012), 3150–3152. https://doi.org/10.1093/bioinformatics/bts565
69. Z. Chen, P. Zhao, F. Li, Y. Wang, A. I. Smith, G. I. Webb, et al., Comprehensive review and assessment of computational methods for predicting RNA post-transcriptional modification sites from RNA sequences, *Brief Bioinf.*, **21** (2019), 1676–1696. https://doi.org/10.1093/bib/bbz112
70. Z. Chen, P. Zhao, C. Li, F. Y. Li, D. X. Xiang, Y. Z. Chen, et al., iLearnPlus: A comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization, *Nucleic Acids Res.*, **49** (2021), 1–19. https://doi.org/10.1093/nar/gkaa1112
71. A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 1–14.
72. S. Ruder, An overview of multi-task learning in deep neural networks, preprint, arXiv: 170605098. https://doi.org/10.48550/arXiv.1706.05098